Methods for the Modeling and Simulation of Sprays and Other Interfacial Flows

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Everett A. Wenzel

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Professor Sean C. Garrick

September, 2019

Acknowledgements

This work was supported in part by the Minnesota Supercomputing Institute, the Abplanalp Sanders Memorial Fellowship, and the University of Minnesota Doctoral Dissertation Fellowship.

Abstract

Interfacial multiphase flows involve the motion of at least two fluids separated by surface tension. Atomizing interfacial flows, colloquially known as sprays, are among the most important fluid dynamic systems because of their ubiquity; power generation, delivery of aerosolized medicines, and productive produce farming all depend fundamentally on the detailed control of sprays. Atomization remains poorly understood because of a historical and persisting inability to accurately and affordably measure the dynamics inside and near the spray orifice outlet – it is therefore desirable to be able to numerically simulate sprays with high fidelity. This dissertation presents computational methods that aim to improve current shortcomings in the modeling and simulation of sprays. Accurately characterizing the interfacial curvature of poorly-resolved liquid structures is addressed by deriving a series of finite particle methods for computing curvature. The methods are verified in analytical curvature tests, and validated against the oscillation frequency of ethanol droplets in air. The finite particle method, leveraging dynamic length scale modification, is demonstrated to out-perform the widely-used height function approach. Tracking the location of interfaces is also addressed, for which a coupled Eulerian-Lagrangian point mass particle scheme is introduced that preserves a well-distributed particle field, can be applied to an arbitrary number of fluids, and does not limit the simulation time step. The Eulerian-Lagrangian method is demonstrated to out-perform contemporary geometric volume of fluid methods at resolutions relevant to spray simulation in a variety of analytical phase tracking tests, and is dynamically evaluated by simulating extending three-phase elliptical regions, droplet dynamics, and Rayleigh-Taylor instabilities. The Eulerian-Lagrangian method is then extended to an approach for consistently and conservatively solving multiphase convection-diffusion problems –

this extension is verified via two analytical heat transfer problems, and robustness is demonstrated by simulating heated air blast atomization. Each of these tests conserves thermal energy and preserves boundedness of the temperature field. This dissertation concludes by outlining paths for consistently and conservatively solving the multiphase Navier-Stokes equations and the multiphase large eddy simulation equations in the coupled Eulerian-Lagrangian point mass particle framework.

Contents

List of Tables

List of Figures

List of Acronyms

ACES	Adjustable curvature evaluation scale
CF	Color function
CSF	Continuum surface force
CSS	Continuum surface stress
DNS	Direct numerical simulation
DPC	Dispersed particle correction
DS	Dynamic stabilization
FPM	Finite particle method
GFM	Ghost fluid method
HF	Height function
LES	Large eddy simulation
LVOF	Lagrangian volume of fluid
MDHF	Mesh decoupled height function
MPI	Message passing interface
MPS	Moving particle semi-implicit
OPS	Optimized particle shifting
PIC	Particle-in-cell
PLIC	Piecewise linear interface calculation
PMP	Point mass particle
PROST	Parabolic reconstruction of surface tension

PS	Particle shifting
RANS	Reynolds averaged Navier-Stokes
RLSG	Refined local surface grid
SGS	Sub-grid scale
SLIPI	Structured laser illumination planar imaging
SPH	Smoothed particle hydrodynamics
VOF	Volume of fluid

Chapter 1

Introduction

1.1 Motivation

Interfacial multiphase flows are fluid dynamic systems that involve at least two immiscible fluids. Familiar examples of interfacial flows include oil spills, splashing ocean waves, and emulsions. Liquid jet atomization (commonly known as spray) is amongst the most important interfacial flows because it determines engine efficiency, exhaust gas composition, properties of particles synthesized via spray drying, and transfer efficiency of fertilizers, herbicides, paints, protective coatings, and medicines [17, 99, 142, 40, 141, 124, 161, 31]. Therefore, technological, environmental, and economic factors motivate the development of accurate and affordable means of predicting atomization dynamics.

Primary atomization is the process by which ligaments and droplets are removed directly from the liquid core of a spray – the liquid core is the contiguous liquid jet expelled from the spray orifice [36]. Primary atomization, while poorly understood, is known to depend on fluid property ratios, long and short wave interface instability, orifice shape, turbulence characteristics in the liquid phase upstream of the orifice, orifice imperfections, and gas phase vortex interactions with the liquid

core [29, 30, 40, 124, 161, 154, 123]. The overall spray structure and initial breakup dynamic depend on the specific spray configuration: spray sheets [23], hollow cone sprays [157], jets in cross flow [109, 51], jets in co-flow [91, 124], and air blast atomization [140] all present different breakup characteristics. Regardless of the type of spray, the dispersed ligaments and droplets resulting from primary atomization continue to breakup after they have separated from the liquid core – a process called secondary atomization. Depending on the application, the droplets may undergo a variety of tertiary processes, including evaporation or deposition (followed then, perhaps, by combustion or solidification). Each of these processes can occur in an open environment, like herbicides sprayed on crops, or in a confined space, like fuel injection into a diesel cylinder [31].

Atomization produces a large number of liquid features with large length-scale ratios. Automotive fuel injection generates approximately 10^7 dispersed droplets and a system length scale ratio of approximately 10^5 [31, 123]. Measuring the atomization process, particularly in the liquid-dense primary atomization region, has been a persistent challenge. The large number of fluid interfaces in the primary atomization region results in multiple scattering of light, precluding the use of standard planar illumination measurement techniques [80, 31]. Structured laser illumination planar imaging (SLIPI) has recently been demonstrated to reduce multiple scattering in moderately dense sprays [70, 6], but has not yet achieved widespread use [31]. Ballistic imaging, another modern spray measurement technique, uses femtosecond lasers and picosecond image gating to mitigate multiple scattering [81, 114], but spatial resolution decreases as optical depth increases [31]. Synchrotron x-ray imaging [64] is capable of measuring many features of the primary atomization region, but is limited in use to only a few research groups because of equipment size and cost [31]. While these new measurement techniques allow for experimental observation

of spray physics with resolution and accuracy exceeding prior capabilities, numerical simulation will continue to contribute significantly to the scientific investigation of spray physics and the practical design of spray systems. Much like experimental investigation, however, the multi-scale, non-linear, multiphase physics of atomization make accurate modeling and simulation a significant challenge.

1.2 Spray simulation

Computational methods used for spray simulation include Reynolds averaged Navier-Stokes simulation (RANS), large eddy simulation (LES), and direct numerical simulation (DNS). RANS, the most affordable approach, solves the Reynolds averaged Navier-Stokes equations, rather than the Navier-Stokes, and relies on linear stability theory and/or stochastic models to treat primary and secondary atomization [115, 102, 122, 4, 41, 118]. The RANS approach fails to accurately describe the atomization process in a predictive manner because it does not resolve transient dynamics [59]. With significant tuning of model parameters, RANS can be used for engineering design in the context of known operating conditions. In contrast to RANS, DNS provides every spatio-temporal detail of the flow field by solving the Navier-Stokes equations. DNS requires orders of magnitude more computational effort than RANS [92, 21, 108, 73, 35, 51, 124, 25, 160]. Only recently has DNS of primary atomization been possible, but the simulations span only a few injector diameters and generally remain under-resolved. The simulations require at least $\mathcal{O}(10^5)$ CPU hours and serve almost exclusively for scientific discovery [40, 123]. Few studies have presented convergence assessments, much less achieved convergence in practical problems [51]. Simulating the entirety of a practical spray process remains impractical, if not impossible, at the present time [123]. One way to reduce the computational cost of simulating a spray is to perform a large eddy simulation (LES) by filtering the Navier-Stokes equations with a low pass filter function, thereby reducing the range

of scales that must be resolved. The filtering operation eliminates nearly all information pertaining to the liquid structures, including droplet size, shape, and relative velocities and temperatures of droplets and gas phase below the filter scale. For this reason, the most common approach for performing LES of atomization is to model the liquid phase as Lagrangian droplets, and the carrier gas phase using standard LES methods [89, 4, 43, 71, 74, 59, 15, 60, 39, 62]. In this framework, droplet motion is governed by a Lagrangian particle equation, such as the Basset-Bousinesq-Oseen equation [60, 61] or a simpler drag law [123], and droplet temperatures and evaporation are treated by model expressions appropriate for the particular application. Primary and secondary atomization are treated by a wide array of deterministic and stochastic breakup models [4, 59, 138, 39, 60, 61]. These Lagrangian treatments perform well when the primary atomization is accurately described (either by prescribing the initial droplet size distribution or by tuning model parameters), but they cannot be used when the primary atomization process is poorly defined. This shortcoming renders the droplet-based spray LES incapable of modeling primary atomization, and therefore sprays, in a predictive manner.

The challenge of performing spray LES is highlighted by the different formulation demands of primary and secondary atomization [161]. Both phases should be treated with the Navier-Stokes equations in the liquid-dense region near the nozzle outlet to accurately predict primary atomization, but this becomes cost-prohibitive in the liquid-sparse regions far from the nozzle [161, 55, 123]. Similarly, Lagrangian treatments of the liquid phase [89, 4, 43, 71, 74, 59, 15, 60, 39, 62] are appropriate when the liquid is dispersed, but cannot predict primary atomization in the dense near-nozzle region [161]. In response to this challenge, researchers have proposed explicit treatment of the primary atomization region with an Eulerian interface capturing method, followed by the transition of atomized droplets to a Lagrangian particle treatment once they become sufficiently small [50, 159]. This model requires either

DNS resolution in the near-nozzle region or closures for all relevant multiphase LES sub-grid scale (SGS) terms. Although these LES SGS terms have been shown to be significant [72, 14], researchers often neglect some or all of the terms when preforming LES of primary atomization, resulting in under-resolved DNS or tacit LES [19, 8, 7, 54, 14, 126, 161, 154, 155, 135]. Researchers have begun developing closures for the SGS interfacial terms, but progress has been slow due to the complexity of the problem. Herrmann and Gorokhovski [55] close the SGS surface tension term by explicitly filtering a fully resolved interface defined by a refined local surface grid. The refined interface is updated by a model equation to account for the interface dynamics, including modeled turbulence and surface tension [55, 49, 53]. Aniszewski et al. [3] and Aniszewski [2] account for the SGS surface tension by performing an approximate deconvolution of either the velocity field or the interface geometry. Efforts in the development of LES models for primary atomization are on-going, and none have, as of the present time, achieved wide acceptance among the research community.

1.3 Overview

This dissertation presents novel methods for use in the modeling and simulation of interfacial flows, each with direct applicability to both DNS and LES of sprays. Each method appears in a dedicated chapter. The contents of the chapters are as follows:

- Chapter 2 presents and evaluates two finite particle method (FPM) approaches for computing curvature of poorly resolved interfaces in the context of volume of fluid simulations. Accurately computing curvature is relevant to spray simulation because surface tension is directly proportional to interfacial curvature. Performance of the FPM methods is evaluated via the L_2 and L_∞ error norms corresponding to the computed curvature of spheres.

- Chapter 3 presents a dynamic evaluation of the FPM through simulations of oscillating ethanol droplets. Performance comparisons are made relative to the widely used height function method. Two FPM implementations with adaptive length scales are presented that provide good dynamic performance across the range of resolutions relevant to spray simulations.

- Chapter 4 presents a coupled Eulerian-Lagrangian point mass particle (PMP) approach for phase tracking in immiscible multiphase flow systems. The Naiver-Stokes equations are solved on a traditional Eulerian grid while the fluid mass and phase information are discretized by Lagrangian particles. The method is novel because the particles move with a velocity that enforces consistency between the particle field density and the fluid density. The governing equations are derived and a numerical method is presented that is applicable to incompressible flows. Phase tracking performance is assessed via standard two-dimensional and three-dimensional phase transport tests as a function of both Eulerian grid resolution and Lagrangian particle resolution. Dynamic performance is assessed via simulations of expanding elliptical regions, stationary and oscillating droplets, a droplet in shear flow, and a Rayleigh-Taylor instability.

- Chapter 5 extends the Eulerian-Lagrangian PMP method to a framework for consistently solving multiphase convection-diffusion problems with discontinuous property variations, presented specifically from the perspective of heat transfer. Performance is assessed with respect to heat diffusion at an air-water interface, convection and diffusion of an initially Gaussian temperature distribution, and the heated air blast atomization of a droplet.

- Chapter 6 proposes a mass-momentum consistent extension of the coupled Eulerian-Lagrangian PMP method for the solution of the Navier-Stokes equations. Consistent transport is required for stable, robust simulation of multiphase flow problems. The governing equations are derived, and a discrete time-integration scheme is introduced. The chapter concludes by introducing a mass-momentum consistent scheme for performing LES of sprays, followed by a discussion of closure requirements.

- Chapter 7 provides a brief summary and conclusions for each chapter, followed by a discussion of on-going and future work.

Chapter 2

Finite particle methods for computing interfacial curvature in volume of fluid simulations

This chapter has been adapted from Wenzel and Garrick [148].

2.1 Introduction

Interfacial flows contain a large range of length scales, variable fluid properties, and nonlinear dynamics including interfacial tension [33]. Sprays, for example, contain curvature scales spanning the injector diameter and the smallest droplet diameter. The ratio of these scales easily exceeds three orders of magnitude, even for the most trivial spray systems. In contemporary spray simulations, and in simulations of the foreseeable future, a large number of simulated droplets are resolved by only a few computational nodes [40]. For example, Herrmann [51] presents the convergence behavior of a round jet in cross flow simulated with the refined level-set grid method.

The highest resolution of the three flow solvers considered discretizes the injector diameter with 64 computational nodes (the refined level-set grid uses twice the flow solver resolution). The primary and secondary atomization structures are therefore resolved by significantly fewer than 64 nodes. Desjardins et al. [22] present two similarly resolved primary atomization simulations, with injector diameters resolved by 51.2 and 100 computational nodes. Under-resolution is naturally exacerbated when other physics are involved, such as phase change and chemical reaction, or when one considers large eddy simulation. While volume of fluid (VOF) transport capabilities have recently improved [47, 152, 13, 104, 106], the challenge of accurately computing interfacial curvature and surface forces across all relevant resolution scales in VOF simulations has not been closed. There is a pressing need for computational schemes capable of accurately describing poorly resolved interfacial geometries [18].

The two most common approaches for computing interfacial geometries in the context of VOF methods are the height function and convolved VOF methods [97, 12]. The height function method provides at least second-order convergence for well-resolved interfaces, but it tends to fail at low resolutions (roughly fewer than 10 grid points across a droplet) [18, 128, 112, 9, 104]. An additional and significant consideration is that convergence is not observed in three-dimensions when the heights are constructed with fewer than nine grid points [105]. The low resolution shortcoming of the height function has been improved by a mesh-decoupled approach [105], but maximum curvature errors measured via the L_∞ error norm remain high ($> 1\%$) for spheres resolved by fewer than about 50 grid points. Other improvements to the low-resolution regime have been made, for example by Bornia et al. [9], but have only been implemented or tested in two-dimensions.

The convolved VOF method computes the interfacial geometry by differentiating the VOF directly with kernel derivatives or by differentiating a convolved form of the VOF with finite differences [12, 1, 117, 153, 18, 34, 135]. The kernel-based schemes

tend to out-perform the finite-difference schemes, but in both cases convergence is only observed when the convolution length is fixed independent of resolution (meaning the number of sampled points increases with resolution) [18]. It has been demonstrated that second-order derivatives acquired via the kernel approach are inaccurate for computationally reasonable support widths [1, 117, 153]. The second order derivatives are accordingly obtained by performing finite-differentiation or first-order kernel differentiation on the first derivatives of the VOF field [1, 117, 153, 34]. This process artificially widens the interface and results in a cascade of derivative operations.

Motivated by the need to accurately compute interfacial tension of poorly-resolved interfacial geometries, we evaluate a finite particle method (FPM) for computing curvature in VOF schemes. The FPM is a Lagrangian framework for simulating fluid flows, but its consistency properties and independence from an underlying mesh make it promising for the computation of geometries on structured and unstructured meshes [84]. We present VOF and convolved VOF implementations that are applicable to Eulerian and Lagrangian frameworks. The present discussion is limited to uniform cartesian meshes to retain a degree of conciseness. Both methods are evaluated via the L_2 and L_∞ error norms corresponding to the curvature of spheres at various resolutions. The performance of the mesh decoupled height function (MDHF) method is also presented for reference and comparison [105]. This chapter concludes with a summary and a brief discussion of implications and conclusions.

2.2 Formulation

2.2.1 Phase representation and curvature

Consider a domain containing two immiscible phases, identified as phase I and phase II. The spatial distribution of phase is mathematically described by the phase-indicating

Heaviside function,

$$\chi(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \text{phase I}, \\ 0, & \mathbf{x} \in \text{phase II}. \end{cases} \tag{2.1}$$

The phase-indicating function is defined at all points in space $\mathbf{x} = \{x^1, x^2, x^3\}$. We define the VOF as the volume fraction of phase I in a control volume j of volume V_j. The VOF is computed from $\chi(\mathbf{x})$ via

$$\psi_j = \frac{1}{V_j} \int_j \chi(\mathbf{x}) dV, \tag{2.2}$$

where $\psi_j = \psi(\mathbf{x}_j)$ is the VOF corresponding to the control volume j with centroid $\mathbf{x}_j = \{x_j^1, x_j^2, x_j^3\}$. The VOF transitions from $\psi_j = 1$ in phase I to $\psi_j = 0$ in phase II over a narrow region of thickness approximately equal to the mesh spacing Δ, where $0 < \psi_j < 1$. Rapid variation in the VOF is not amenable to numerical differentiation [1], so we introduce a "smooth" representation of phase termed the color function ϕ. This color function transitions from $\phi = 1$ in phase I to $\phi = 0$ in phase II over an interfacial thickness that is greater than Δ. The value of the color function at point $\mathbf{x}_j$ is obtained by convolving the VOF over N_p points near $\mathbf{x}_j$:

$$\phi_j = \frac{\sum_{i=1}^{N_p} \psi_i W_1(\mathbf{x}_i - \mathbf{x}_j, h_1) V_i}{\sum_{i=1}^{N_p} W_1(\mathbf{x}_i - \mathbf{x}_j, h_1) V_i}. \tag{2.3}$$

In the above expression, W_1 is a convolution or weight function with a compact influence radius h_1. The number of points sampled in the convolution is the number of points on the computational mesh that fall within a distance h_1 of point $\mathbf{x}_j$. It is possible to generate a variety of equally valid color functions by changing W_1 or h_1. For the purposes of computing curvature, W_1 and h_1 are chosen so that the first and second-order spatial derivatives of ϕ are well defined. The expression for curvature used in this work is

$$\kappa = \frac{\nabla\phi H \nabla\phi^T - |\nabla\phi|^2 \,\text{Trace}(H)}{2|\nabla\phi|^3}, \tag{2.4}$$

where H is the Hessian, $H = \nabla(\nabla\phi)$ [38].

2.2.2 Derivatives of the color function

We have devised a process, in the formalism of the FPM [84], that facilitates computation of derivatives of ϕ. Once acquired, these derivatives are used to compute the curvature κ via Eq. (2.4). A second-order Taylor series expansion relates the value of the color function ϕ at nearby locations $\mathbf{x} = \{x^1, x^2, x^3\}$ and $\mathbf{x}_o = \{x_o^1, x_o^2, x_o^3\}$,

$$\phi(\mathbf{x}) = \phi(\mathbf{x}_o) + (x^\alpha - x_o^\alpha)\frac{\partial\phi(\mathbf{x}_o)}{\partial x^\alpha} + \frac{1}{2}(x^\alpha - x_o^\alpha)(x^\beta - x_o^\beta)\frac{\partial^2\phi(\mathbf{x}_o)}{\partial x^\alpha \partial x^\beta} + \mathcal{O}\left[(\mathbf{x} - \mathbf{x}_o)^3\right], \quad (2.5)$$

where the superscripts denote a component of the position vector $\mathbf{x}$ and repeated superscript indices imply summation. We multiply Eq. (2.5) by a limited support weight function $W_2(\mathbf{x} - \mathbf{x}_o, h_2)$, integrate over the volume of limited support, and then approximate the integral by summation over discrete elements j of volume V_j. This results in a spatially local and weighted relation between $\phi_o = \phi(\mathbf{x}_o)$ and $\phi_j = \phi(\mathbf{x}_j)$, where $\mathbf{x}_j$ is the set of N_p points within h_2 of $\mathbf{x}_o$,

$$\begin{aligned}
\sum_{j=1}^{N_p} \phi_j W_2 V_j &= \phi_o \sum_{j=1}^{N_p} W_2 V_j + \frac{\partial\phi_o}{\partial x^\alpha} \sum_{j=1}^{N_p} \left(x_j^\alpha - x_o^\alpha\right) W_2 V_j \\
&+ \frac{\partial^2\phi_o}{\partial x^\alpha \partial x^\beta} \sum_{j=1}^{N_p} \frac{1}{2}(x_j^\alpha - x_o^\alpha)(x_j^\beta - x_o^\beta) W_2 V_j \\
&+ \sum_{j=1}^{N_p} \mathcal{O}\left[(\mathbf{x}_j - \mathbf{x}_o)^3\right] W_2 V_j. \qquad (2.6)
\end{aligned}$$

The arguments of W_2 have been dropped for conciseness. Ten unknowns, ϕ_o and its derivatives, appear in Eq. (2.6). We follow the standard FPM procedure of introducing additional weight functions that are spatial derivatives of the base weight function

W_2 [84]. The result is a three-dimensional sampling basis Ω,

$$\Omega = \left[W_2 \; \frac{\partial W_2}{\partial x^1} \; \frac{\partial W_2}{\partial x^2} \; \frac{\partial W_2}{\partial x^3} \; \frac{\partial^2 W_2}{\partial x^1 \partial x^1} \; \frac{\partial^2 W_2}{\partial x^2 \partial x^2} \; \frac{\partial^2 W_2}{\partial x^3 \partial x^3} \; \frac{\partial^2 W_2}{\partial x^1 \partial x^2} \; \frac{\partial^2 W_2}{\partial x^1 \partial x^3} \; \frac{\partial^2 W_2}{\partial x^2 \partial x^3} \right]^T . \quad (2.7)$$

Using the sampling basis to construct the equivalent of Eq. (2.6) for each weight in Ω and dropping terms of order three returns the FPM system,

$$\begin{aligned} \sum_{j=1}^{N_p} \phi_j \Omega^k V_j &= \phi_o \sum_{j=1}^{N_p} \Omega^k V_j + \frac{\partial \phi_o}{\partial x^\alpha} \sum_{j=1}^{N_p} \left(x_j^\alpha - x_o^\alpha \right) \Omega^k V_j \\ &+ \frac{\partial^2 \phi_o}{\partial x^\alpha \partial x^\beta} \sum_{j=1}^{N_p} \frac{1}{2} (x_j^\alpha - x_o^\alpha)(x_j^\beta - x_o^\beta) \Omega^k V_j . \end{aligned} \quad (2.8)$$

where the index k ranges from 1 to 10. Solving the system of equations defined by Eq. (2.8) returns the color function and its first and second derivatives at location x_o (ϕ_o, $\partial \phi_o / \partial x^\alpha$, and $\partial^2 \phi_o / \partial x^\alpha \partial x^\beta$).

2.2.3 Curvature from the VOF

The computational cost associated with the solution of Eq. (2.3) may be avoided by computing the curvature from the VOF directly. The first step in arriving at such a methodology is to add $\psi(\mathbf{x}) - \psi_o$ to both sides of Eq. (2.5), yielding (after minor manipulation)

$$\begin{aligned} \psi(\mathbf{x}) - \psi_o &= (x^\alpha - x_o^\alpha) \frac{\partial \phi_o}{\partial x^\alpha} + \frac{1}{2} (x^\alpha - x_o^\alpha)(x^\beta - x_o^\beta) \frac{\partial^2 \phi_o}{\partial x^\alpha \partial x^\beta} \\ &- [\phi(\mathbf{x}) - \phi_o] + [\psi(\mathbf{x}) - \psi_o] + \mathcal{O}\left[(\mathbf{x} - \mathbf{x}_o)^3 \right] . \end{aligned} \quad (2.9)$$

We multiply Eq. (2.9) by $\partial W_2/\partial x^1$ and integrate to obtain (after dropping third order terms)

$$\begin{aligned}\sum_{j=1}^{N_p}(\psi_j-\psi_o)\frac{\partial W_2}{\partial x^1}V_j &= \frac{\partial\phi_o}{\partial x^\alpha}\sum_{j=1}^{N_p}\left(x_j^\alpha-x_o^\alpha\right)\frac{\partial W_2}{\partial x^1}V_j \\ &+ \frac{\partial^2\phi_o}{\partial x^\alpha\partial x^\beta}\sum_{j=1}^{N_p}\frac{1}{2}(x_j^\alpha-x_o^\alpha)(x_j^\beta-x_o^\beta)\frac{\partial W_2}{\partial x^1}V_j \\ &- \sum_{j=1}^{N_p}\left[\phi_j-\phi_o\right]\frac{\partial W_2}{\partial x^1}V_j+\sum_{j=1}^{N_p}\left[\psi_j-\psi_o\right]\frac{\partial W_2}{\partial x^1}V_j. \quad (2.10)\end{aligned}$$

The left-hand-side of Eq. (2.10) is the kernel approximation of $\partial\psi_o/\partial x^1$,

$$\frac{\partial\psi_o}{\partial x^1}\approx\sum_{j=1}^{N_p}(\psi_j-\psi_o)\frac{\partial W_2}{\partial x^1}V_j, \quad (2.11)$$

and the third term on the right-hand-side is the kernel approximation of $\partial\phi_o/\partial x^1$

$$\frac{\partial\phi_o}{\partial x^1}\approx\sum_{j=1}^{N_p}\left[\phi_j-\phi_o\right]\frac{\partial W_2}{\partial x^1}V_j. \quad (2.12)$$

We make use of these approximations to re-write Eq. (2.10) as

$$\begin{aligned}\frac{\partial\psi_o}{\partial x^1} &= \frac{\partial\phi_o}{\partial x^\alpha}\sum_{j=1}^{N_p}\left(x_j^\alpha-x_o^\alpha\right)\frac{\partial W_2}{\partial x^1}V_j+\frac{\partial^2\phi_o}{\partial x^\alpha\partial x^\beta}\sum_{j=1}^{N_p}\frac{1}{2}(x_j^\alpha-x_o^\alpha)(x_j^\beta-x_o^\beta)\frac{\partial W_2}{\partial x^1}V_j \\ &+ \frac{\partial\psi_o}{\partial x^1}-\frac{\partial\phi_o}{\partial x^1}. \quad (2.13)\end{aligned}$$

Both kernel expressions, $\partial\psi_o/\partial x^1$ and $\partial\phi_o/\partial x^1$, are at least twice differentiable provided W_2 is at least thrice differentiable [153]. Because all of the terms in Eq. (2.13)

are differentiable, we assume that a different color function $\varphi(\mathbf{x}_o)$ exists that satisfies

$$\frac{\partial\varphi_o}{\partial x^\alpha}\sum_{j=1}^{N_p}\left(x_j^\alpha - x_o^\alpha\right)\frac{\partial W_2}{\partial x^1}V_j + \frac{\partial^2\varphi_o}{\partial x^\alpha\partial x^\beta}\sum_{j=1}^{N_p}\frac{1}{2}(x_j^\alpha - x_o^\alpha)(x_j^\beta - x_o^\beta)\frac{\partial W_2}{\partial x^1}V_j =$$

$$\frac{\partial\phi_o}{\partial x^\alpha}\sum_{j=1}^{N_p}\left(x_j^\alpha - x_o^\alpha\right)\frac{\partial W_2}{\partial x^1}V_j + \frac{\partial^2\phi_o}{\partial x^\alpha\partial x^\beta}\sum_{j=1}^{N_p}\frac{1}{2}(x_j^\alpha - x_o^\alpha)(x_j^\beta - x_o^\beta)\frac{\partial W_2}{\partial x^1}V_j + \frac{\partial\psi_o}{\partial x^1} - \frac{\partial\phi_o}{\partial x^1}. \tag{2.14}$$

Substituting this expression into Eq. (2.13) returns

$$\begin{aligned}\frac{\partial\psi_o}{\partial x^1} &= \frac{\partial\varphi_o}{\partial x^\alpha}\sum_{j=1}^{N_p}\left(x_j^\alpha - x_o^\alpha\right)\frac{\partial W_2}{\partial x^1}V_j \\ &+ \frac{\partial^2\varphi_o}{\partial x^\alpha\partial x^\beta}\sum_{j=1}^{N_p}\frac{1}{2}(x_j^\alpha - x_o^\alpha)(x_j^\beta - x_o^\beta)\frac{\partial W_2}{\partial x^1}V_j.\end{aligned} \tag{2.15}$$

Nine unknowns (the derivatives of φ_o) appear in Eq. (2.15), as opposed to the ten unknowns in Eq. (2.8). We therefore implement a sampling basis with nine weights,

$$\Omega = \left[\frac{\partial W_2}{\partial x^1}\ \frac{\partial W_2}{\partial x^2}\ \frac{\partial W_2}{\partial x^3}\ \frac{\partial^2 W_2}{\partial x^1\partial x^1}\ \frac{\partial^2 W_2}{\partial x^2\partial x^2}\ \frac{\partial^2 W_2}{\partial x^3\partial x^3}\ \frac{\partial^2 W_2}{\partial x^1\partial x^2}\ \frac{\partial^2 W_2}{\partial x^1\partial x^3}\ \frac{\partial^2 W_2}{\partial x^2\partial x^3}\right]^T, \tag{2.16}$$

under the assumption that

$$\begin{aligned}\frac{\partial\varphi_o}{\partial x^\alpha}\sum_{j=1}^{N_p}\left(x_j^\alpha - x_o^\alpha\right)\Omega^k V_j + \frac{\partial^2\varphi_o}{\partial x^\alpha\partial x^\beta}\sum_{j=1}^{N_p}\frac{1}{2}(x_j^\alpha - x_o^\alpha)(x_j^\beta - x_o^\beta)\Omega^k V_j &= \\ \frac{\partial\phi_o}{\partial x^\alpha}\sum_{j=1}^{N_p}\left(x_j^\alpha - x_o^\alpha\right)\Omega^k V_j + \frac{\partial^2\phi_o}{\partial x^\alpha\partial x^\beta}\sum_{j=1}^{N_p}\frac{1}{2}(x_j^\alpha - x_o^\alpha)(x_j^\beta - x_o^\beta)\Omega^k V_j &- \\ \sum_{j=1}^{N_p}\left[\phi_j - \phi_o\right]\Omega^k V_j + \sum_{j=1}^{N_p}\left[\psi_j - \psi_o\right]\Omega^k V_j,\end{aligned} \tag{2.17}$$

and use the basis to construct the corresponding FPM system,

$$\sum_{j=1}^{N_p} (\psi_j - \psi_o)\, \Omega^k V_j = \frac{\partial \varphi_o}{\partial x^\alpha} \sum_{j=1}^{N_p} \left(x_j^\alpha - x_o^\alpha\right) \Omega^k V_j + \frac{\partial^2 \varphi_o}{\partial x^\alpha \partial x^\beta} \sum_{j=1}^{N_p} \frac{1}{2} (x_j^\alpha - x_o^\alpha)(x_j^\beta - x_o^\beta) \Omega^k V_j, \tag{2.18}$$

where the index k ranges from 1 to 9. Solving the system of equations defined by Eq. (2.18) returns $\partial \varphi_o / \partial x^\alpha$ and $\partial^2 \varphi_o / \partial x^\alpha \partial x^\beta$. Substituting these spatial derivatives of φ_o for the spatial derivatives of ϕ in Eq. (2.4) returns the curvature κ at point $\mathbf{x}_o$.

The left-hand-side of Eq. (2.18) is a column vector containing the kernel approximations of the nine spatial derivatives of the VOF ψ evaluated at position $\mathbf{x}_o$. These kernel derivatives are associated with the interfacial length scales and the influence radius h_2 of the sampling basis Ω. The kernel approximations have an implicit smoothness due to the sampling of ψ with Ω. Therefore, while the VOF ψ may not be smooth on the Eulerian mesh, the kernel approximations of its derivatives used in Eq. (2.18) are all smooth, provided h_2 is sufficiently large.

2.2.4 Implementation optimizations

For calculations performed on time-independent grids, computing and storing constants at the beginning of the simulation considerably reduces computational costs. This requires writing the FPM system, chosen for this demonstration as Eq. (2.8), at point $\mathbf{x}_o$ in matrix form,

$$\mathbf{A}\mathbf{b} = \mathbf{c}, \tag{2.19}$$

where $\mathbf{A}$ is the coefficient matrix,

$$\mathbf{A} = \begin{bmatrix} \sum_{j=1}^{N_p} \Omega^1 V_j & \sum_{j=1}^{N_p} \left(x_j^1 - x_o^1\right) \Omega^1 V_j & \dots & \sum_{j=1}^{N_p} \frac{1}{2}(x_j^2 - x_o^2)(x_j^3 - x_o^3)\Omega^1 V_j \\ \sum_{j=1}^{N_p} \Omega^2 V_j & \sum_{j=1}^{N_p} \left(x_j^1 - x_o^1\right) \Omega^2 V_j & \dots & \sum_{j=1}^{N_p} \frac{1}{2}(x_j^2 - x_o^2)(x_j^3 - x_o^3)\Omega^2 V_j \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^{N_p} \Omega^{10} V_j & \sum_{j=1}^{N_p} \left(x_j^1 - x_o^1\right) \Omega^{10} V_j & \dots & \sum_{j=1}^{N_p} \frac{1}{2}(x_j^2 - x_o^2)(x_j^3 - x_o^3)\Omega^{10} V_j \end{bmatrix}, \tag{2.20}$$

$\mathbf{b}$ is the solution vector,

$$\mathbf{b} = \begin{bmatrix} \phi_o \\ \frac{\partial \phi_o}{\partial x_1} \\ \vdots \\ \frac{\partial^2 \phi_o}{\partial x_2 \partial x_3} \end{bmatrix}, \tag{2.21}$$

and $\mathbf{c}$ is the weighted color function vector,

$$\mathbf{c} = \begin{bmatrix} \sum_{j=1}^{N_p} \phi_j \Omega^1 V_j \\ \sum_{j=1}^{N_p} \phi_j \Omega^2 V_j \\ \vdots \\ \sum_{j=1}^{N_p} \phi_j \Omega^{10} V_j \end{bmatrix}. \tag{2.22}$$

Solution of the FPM system is then defined by

$$\mathbf{b} = \mathbf{A}^{-1}\mathbf{c}. \tag{2.23}$$

The inverse of the coefficient matrix $\mathbf{A}^{-1}$ can be computed and stored for time-independent grids ($\mathbf{x}_o$ and $\mathbf{x}_j$ do not change in time). On a uniform cartesian grid, $\mathbf{A}^{-1}$ is the same at every grid point. On an unstructured or non-uniform grid, $\mathbf{A}^{-1}$ is unique for every grid point. One can also compute and store $\Omega^k V_j$ which reduces the runtime cost of computing the $\mathbf{c}$ vector. With these optimizations, solution of the FPM system becomes a two-step process. First, the $\mathbf{c}$ vector is completed by multiplying the stored values of $\Omega^k V_j$ by ϕ_j. Next, the $\mathbf{c}$ vector is multiplied by the stored inverse of the coefficient matrix $\mathbf{A}^{-1}$. This two-step process returns the spatial

derivatives of ϕ_o in the solution vector $\mathbf{b}$. These derivatives are used to compute curvature via Eq. (2.4).

2.3 Performance assessment

2.3.1 Evaluation methodology

The accuracy of the proposed methods is assessed by comparing the computed curvature of spheres to the analytical curvature. In order to perform these evaluations, we approximately recreate the procedure used to evaluate the MDHF method [105]. Two-dimensional geometries are not considered because most applications of practical interest are three-dimensional.

A sphere of diameter $D = 0.4$ is placed in a cube of side length $L = 1$. A set of $\mathcal{N}$ equally spaced points are generated on the surface of the sphere. A sufficient number of points are generated such that there is at least one point in each computational cell intersected by the surface of the sphere. The predicted curvature κ is compared to the analytical curvature κ_a at these $\mathcal{N}$ points via the L_2 and L_∞ norms defined by [18, 105]

$$L_2 = \sqrt{\frac{\sum_{\mathcal{N}} (\kappa - \kappa_a)^2}{\sum_{\mathcal{N}} \kappa_a^2}} \tag{2.24}$$

and

$$L_\infty = \max \left| \frac{\kappa - \kappa_a}{\kappa_a} \right|. \tag{2.25}$$

The L_2 norm is averaged over all simulations per mesh resolution, while the L_∞ norm is taken as the maximum over all simulations.

The proposed evaluation methodology neglects the contribution of error from uncertainty in the precise interface location; this error is neglected by locating the $\mathcal{N}$ points identically on the interface. Neglecting this error is intentional because it isolates the errors due to the FPM expressions. In Section 2.4 we explore the effects of

interface location error by defining the interface with the piecewise linear interface calculation (PLIC) method [119].

2.3.2 Numerical parameters

The domain is discretized by a uniform mesh ranging in resolution from 6^3 to 129^3 computational nodes. Padded regions are added when necessary to accommodate stencils that exceed the domain boundary. The uniform mesh spacing Δ ranges from $\Delta = 1/5$ to $\Delta = 1/128$. For each mesh resolution, five simulations are performed where the center of the sphere is displaced from the center of the cube in the x^1, x^2, and x^3 directions by random numbers ranging from $-\Delta$ to Δ. The perturbations produce different alignments between the sphere and the mesh and therefore result in different errors.

The VOF is initialized using an approach similar to that used in [18]. Each control volume is divided into a $3 \times 3 \times 3$ sub-mesh. If a control volume in the sub-mesh contains the interface, it is further divided into a $33 \times 33 \times 33$ mesh. The mesh spacing of the finest mesh is therefore $\Delta/99$. If a control volume in the finest mesh contains the interface, the VOF in the control volume is computed by assuming the interface is a plane with a normal vector corresponding to the $D = 0.4$ sphere. The VOF corresponding to the original control volume is then computed by volume averaging all VOF values contained within its bounds. As a form of verification, the VOF was also initialized with a Monte Carlo method. No significant change in the curvature results are observed for the different VOF initialization approaches.

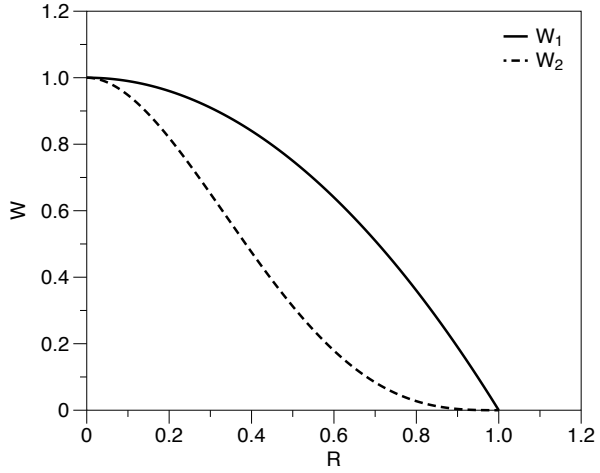

Figure 2.1. Profiles of weight functions W_1 and W_2.

2.3.3 Weight functions

The color function-based computations are performed by initially convolving the VOF according to Eq. (2.3). The weight function used for this operation is defined as

$$W_1(R, h_1) = \begin{cases} (1 - R^2), & 0 \leq R \leq 1, \\ 0, & R > 1, \end{cases} \tag{2.26}$$

where $R = |\mathbf{x}_j - \mathbf{x}_o| / h_1$. The weight function used to construct the FPM sampling basis Ω for both the color function and VOF-based methods is taken as

$$W_2(R, h_2) = \begin{cases} (-3R^4 + 8R^3 - 6R^2 + 1), & 0 \leq R \leq 1, \\ 0, & R > 1, \end{cases} \tag{2.27}$$

where $R = |\mathbf{x}_j - \mathbf{x}_o| / h_2$ [87]. Weights W_1 and W_2 are provided for reference in Fig. 2.1.

Smaller influence radii are associated with lower computational costs and potentially lower accuracy. For example, if an influence radius is selected as $h = 2\Delta$, the weight function is non-zero at the centroid of $N_p = 27$ neighboring nodes (on a

uniform grid). This may or may not be sufficient to approximate the sampled function. The number of points required for a high-fidelity approximation depends on the function being sampled. Small influence radii are sometimes required because the method assumes that one contiguous interface is sampled – this is also true for MDHF, convolved VOF, and level-set based methods [48]. Sampling more than one interface introduces an error, so small influence radii should be applied to flows with multiple, close proximity interfaces. The multiple interface error can be eliminated by temporarily removing all nearby interfaces disconnected from the interface of interest [88, 129], but this remedy adds implementation complexity. In the following results, we limit the discussion to reasonably small influence radii.

The number of grid points sampled by a weight function with influence radius h depends partly on whether the calculations are interface-centered ($\mathbf{x}_o$ falls on the interface) or cell-centered ($\mathbf{x}_o$ falls on a grid point). This is demonstrated by considering an influence radius that is an integer multiple of the mesh spacing, for example $h = 3\Delta$. On a Cartesian mesh with cell-centered calculations, the only relevant cells are those within $h = 2\Delta$ of $\mathbf{x}_o$ in each of the orthogonal directions. This is because the weight function reaches a magnitude of zero when the separation between the sampling point and the origin of the calculation $\mathbf{x}_o$ equals the influence radius. In the case of cell-centered calculations on a uniform mesh, the support width is best described by an effective influence radius,

$$h^\star = h - \Delta. \tag{2.28}$$

On unstructured or curvilinear meshes, or when the calculations are interface-centered, the effective influence radius and the influence radius are equal, $h^\star = h$, because the effective support depends on the local mesh and its alignment with the interface. In other words, the separation between sampled points is irregular, so the entire influence radius must be considered. In the following sections we present all results as a function of the effective influence radius $h^\star$.

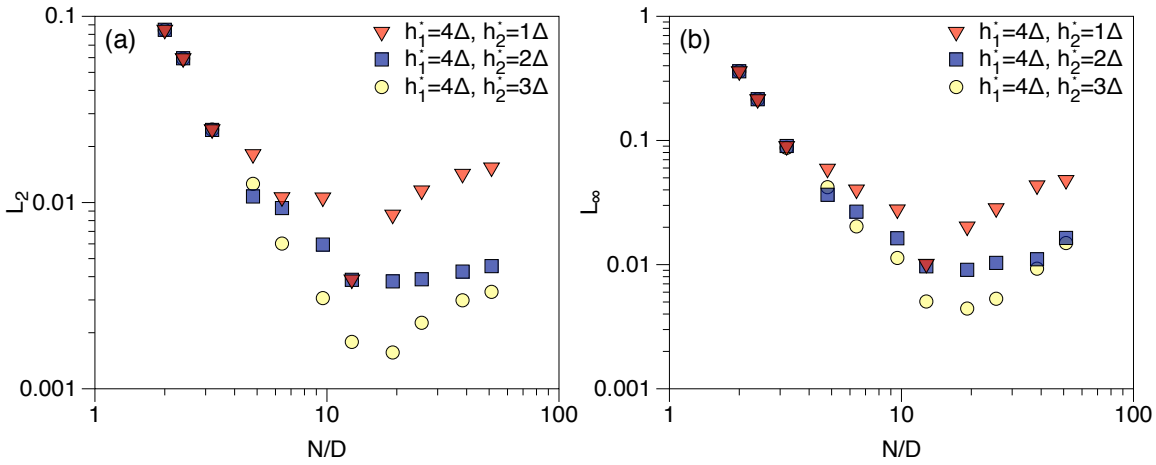

Figure 2.2. Error norms for cell-centered calculations of curvature from the color function: (a) L_2 error norms; (b) L_∞ error norms.

2.3.4 Curvature from the color function

Two methods for computing the interfacial derivatives are considered. In the first, all computations are performed at cell centers – the VOF is initialized at the cell-centers, the color function is computed at the cell-centers, and the curvature is computed at the cell-centers. The cell-centered curvatures are then tri-linearly interpolated to the $\mathcal{N}$ interfacial points. This method allows for the pre-calculation and storage techniques discussed in Section 2.4 because the locations of $\mathbf{x}_o$ and $\mathbf{x}_j$ are fixed at the cell-centers. In the second case, the curvature is computed at the $\mathcal{N}$ interfacial points – the VOF is initialized at the cell-centers, the color function is computed at the cell-centers, and the curvature is computed at the $\mathcal{N}$ interfacial points. The second method does not allow for pre-calculation and storage because the location of $\mathbf{x}_o$ moves with the interface. The distinction between these two methods is made because they provide different sources of error.

Curvature errors from cell-centered calculations based on the color function are shown in Fig. 2.2. Results are obtained for an influence radius of $h_1^\star = 4\Delta$, and three

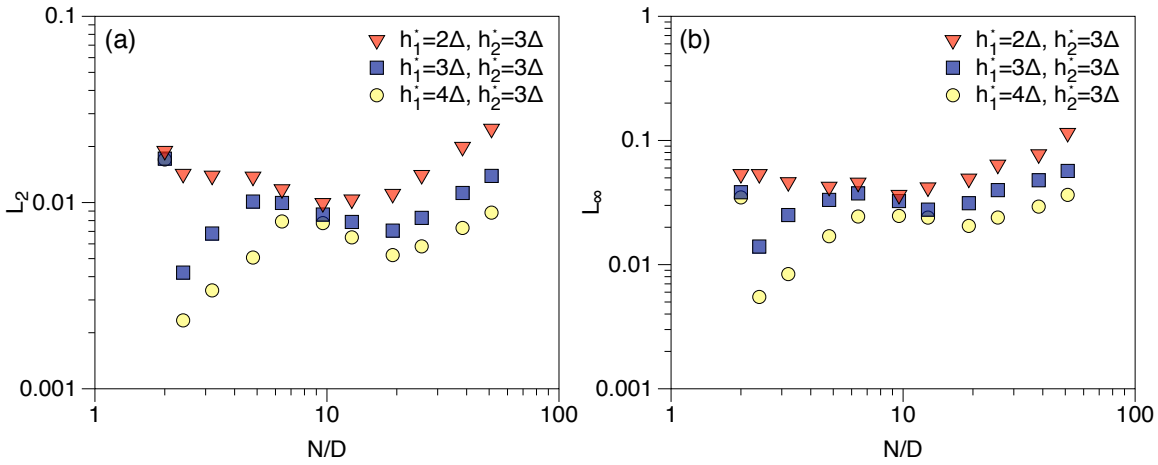

Figure 2.3. Error norms for interface-centered calculations of curvature from the color function: (a) L_2 error norms; (b) L_∞ error norms.

values of $h_2^\star$ ($h_2^\star = \Delta$, $h_2^\star = 2\Delta$ and $h_2^\star = 3\Delta$). The errors are presented as a function of the number of computational nodes N across the sphere diameter D, denoted by N/D. The errors generally decrease as the resolution increases. However, the errors begin to increase beyond a resolution of $N/D = 20$. Similar to the standard convolved VOF method, convergence is not observed. Lack of convergence is attributed to high-frequency variations in the VOF field [18]. For $N/D \leq 3$, all values of $h_2^\star$ yield the same result. When the interface is poorly resolved, interpolation does not appropriately characterize the sub-grid-scale curvature and dominates the error. Beyond resolutions of $N/D = 3$, the solution becomes more accurate as $h_2^\star$ increases. For $N/D > 20$, the solution becomes less accurate due to increased influence of the high frequency VOF variations. Despite the increase in error, the L_2 norm is less than 1% for droplets resolved by greater than 5Δ for $h_2^\star > \Delta$ (up to the maximum tested value of $N/D = 51.2$). The L_∞ norm is larger than L_2 for all cases, as it represents the global maximum error of all five simulations.

Results for the interface-centered calculations are shown in Fig. 2.3. Calculations are performed for an FPM influence radius of $h_2^\star = 3\Delta$, and three values of $h_1^\star$ ($h_1^\star =$

2Δ, $h_1^\star = 3\Delta$ and $h_1^\star = 4\Delta$). The figure reveals that the interface-centered calculations produce smaller errors than the cell-centered calculations for the lowest resolutions (approximately $N/D \leq 5$). Both L_2 and L_∞ errors decrease with increasing $h_1^\star$. However, they are relatively insensitive to resolution – the errors do not depend strongly on N/D over the range tested. For $h_1^\star = 4\Delta$ and $h_2^\star = 3\Delta$, the L_2 error is less than 1% for all resolutions tested except for the lowest of $N/D = 2$. For the same parameters, the L_∞ error is less than 4% for all resolutions. Sufficient smoothing of the VOF is required to obtain results that do not quickly diverge, and the results suggest $h_1^\star = 2\Delta$ provides insufficient smoothing.

2.3.5 Modified color function approach

There are multiple variants of the color function approach. We limit the discussion of variants to one simple modification that reduces the computational cost of computing curvature from the color function.

If ϕ_o is obtained via Eq. (2.3), the first term on the right-hand-side of Eq. (2.8) can be subtracted from both sides of the equation, resulting in a new expression,

$$\sum_{j=1}^{N_p} (\phi_j - \phi_o)\, \Omega^k V_j = \frac{\partial \phi_o}{\partial \mathbf{x}^\alpha} \sum_{j=1}^{N_p} \left(\mathbf{x}_j^\alpha - \mathbf{x}_o^\alpha\right) \Omega^k V_j + \frac{\partial^2 \phi_o}{\partial \mathbf{x}^\alpha \partial \mathbf{x}^\beta} \sum_{j=1}^{N_p} \frac{1}{2} (\mathbf{x}_j^\alpha - \mathbf{x}_o^\alpha)(\mathbf{x}_j^\beta - \mathbf{x}_o^\beta) \Omega^k V_j. \tag{2.29}$$

The modified FPM expression in Eq. (2.29) is a system of nine, rather than ten equations like Eq. (2.8), because the value of ϕ_o is known. The sampling basis Ω defined by Eq. (2.7) can therefore be reduced in size from ten to nine elements by removing W_2. The potential benefit of Eq. (2.29) is that it requires the population and solution of a 9×9 matrix rather than a 10×10 matrix, and therefore reduces computational cost by approximately 20%.

A comparison of the cell-centered results produced by Eq. (2.8) and Eq. (2.29) appears in Fig. 2.4. For both cases, $h_1^\star = 4\Delta$ and $h_2^\star = 3\Delta$. Performance with respect

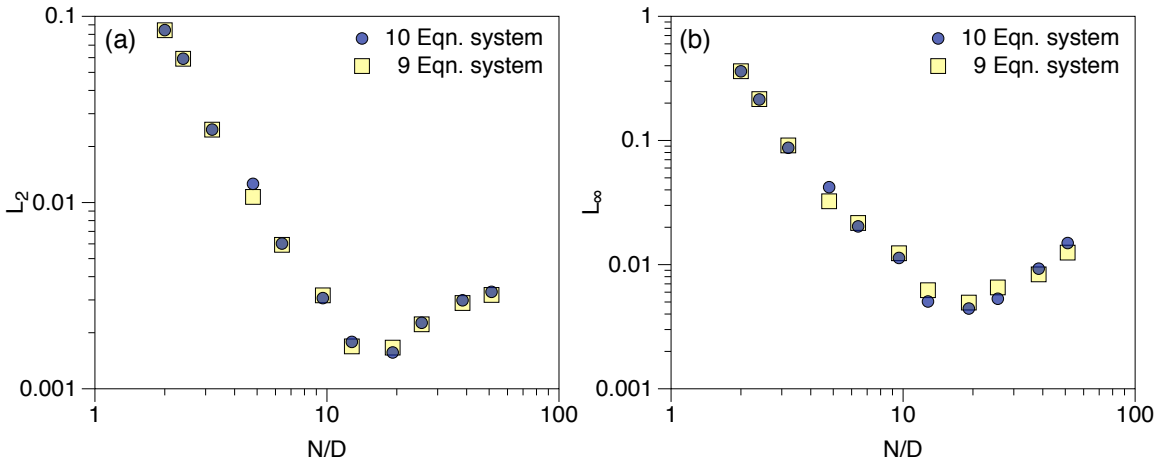

Figure 2.4. Comparison of errors produced by the 10 equation system of Eq. (2.8) and the 9 equation system of Eq. (2.29) for $h_1^\star = 4\Delta$ and $h_2^\star = 3\Delta$: (a) L_2 error norms; (b) L_∞ error norms.

to both L_2 and L_∞ is comparable over the range of N/D considered, suggesting the constraint on ϕ_o required to derive Eq. (2.29) does not strongly impact the solution. The similar performances of Eq. (2.8) and Eq. (2.29) suggest the use of Eq. (2.29) is favorable, considering the reduction in computational expense.

2.3.6 Curvature from the VOF

Only cell-centered curvatures are considered for the VOF-based calculations. Curvature errors obtained by performing VOF-based calculations are shown in Fig. 2.5. The trends observed in the error are similar to those in the color function-based curvature. For small $h_2^\star$, such as $h_2^\star = 2\Delta$, poor performance is observed. For $h_2^\star > 3\Delta$, the L_2 errors are effectively less than 1% for resolutions $5 < N/D < 25$. For $h_2^\star = 5$, the L_2 errors are less than 1% for all tested resolutions greater than 5 N/D. A maximum characteristic error of $L_2 \approx 9.25\%$ is achieved for spheres resolved by two nodes. The color function and VOF methods differ in the relative magnitude of error. For a given $h_2^\star$, the errors in the VOF-obtained curvature are between two and ten times greater

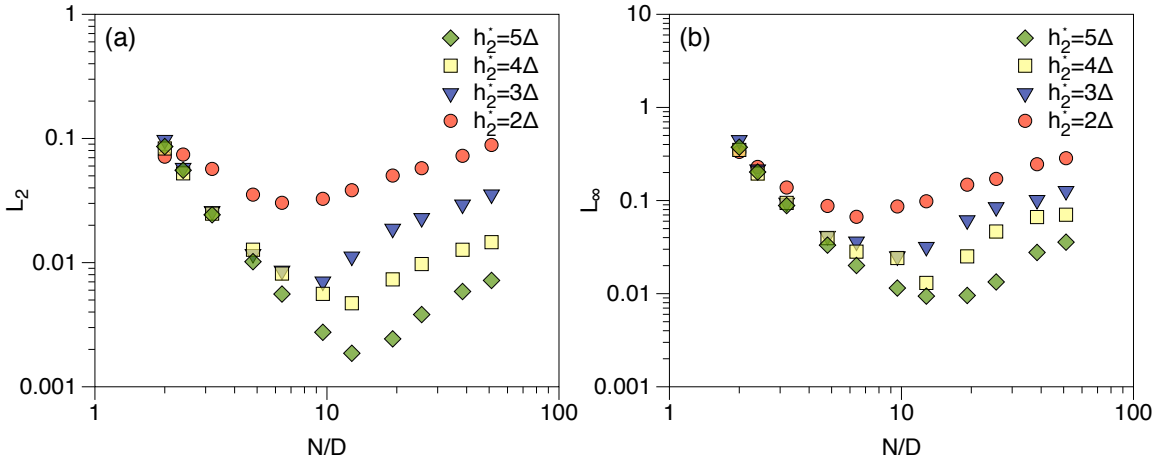

Figure 2.5. Error norms for cell-centered calculations of curvature from the VOF: (a) L_2 error norms; (b) L_∞ error norms.

Table 2.1. Parameters for hybrid FPM #1 in Fig. 2.6.

Resolution	Interface/Cell-centered	$h_1^\star$	$h_2^\star$
$N/D < 5$	Interface	4Δ	3Δ
$5 \leq N/D < 23$	Cell	4Δ	3Δ
$23 \leq N/D < 35$	Cell	6Δ	3Δ
$35 \leq N/D < 50$	Cell	7Δ	3Δ
$N/D \geq 50$	Cell	11Δ	3Δ

than the curvature obtained from the color function.

2.3.7 Hybrid computations

Accuracy depends on the resolution, the influence radii, and whether the calculations are interface-centered or cell-centered. A hybrid approach that changes as a function of N/D can optimize accuracy. One such hybrid approach is defined in Table 2.1. It performs color function-based calculations that are interface-centered at low resolutions and cell-centered at high resolutions. Smoothness of the color function is improved by increasing $h_1^\star$ with increasing N/D. Performance of the hybrid FPM

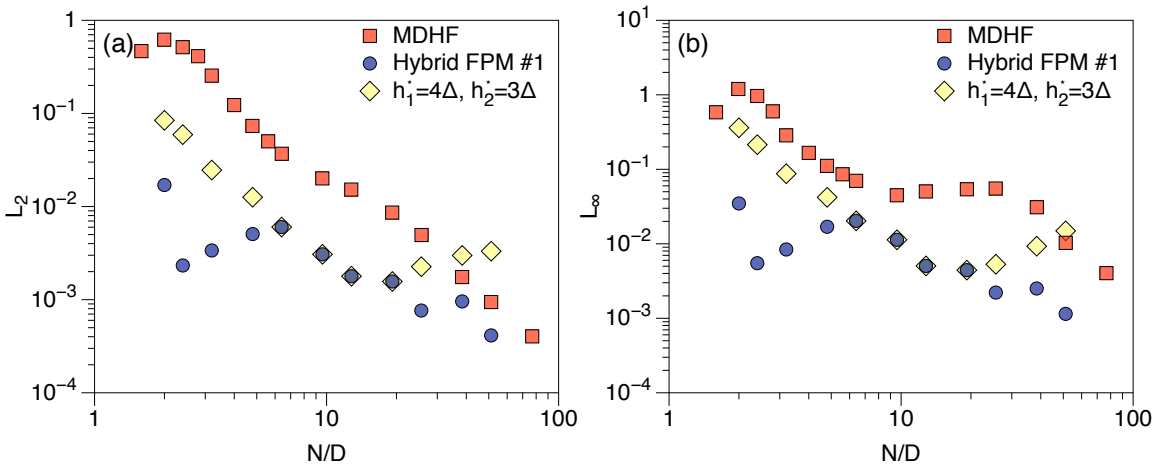

Figure 2.6. Performance comparison of hybrid FPM #1 (defined in Table 2.1), the color function-based FPM with cell-centered calculations, and the MDHF method: (a) L_2 error norms; (b) L_∞ error norms. MDHF results are taken from [105].

is demonstrated in Fig. 2.6. For comparison, Fig. 2.6 also includes the cell-centered approach with curvature computed from the color function ($h_1^\star = 4\Delta$ and $h_2^\star = 3\Delta$), and the MDHF results from [105]. (These results were electronically recovered and serve as reference only.) The MDHF uses nine nodes to construct the heights and uses second-order differencing.

Figure 2.6 shows that the L_2 and L_∞ errors decrease with increasing resolution for the hybrid FPM. Increasing the color function influence radius, $h_1^\star$, as a function of N/D damps high frequency variations in the VOF and allows for high accuracy at large N/D. The results in Fig. 2.6 demonstrate that the FPM approach has the ability to out-perform the MDHF with respect to both the L_2 and L_∞ errors across the range of resolutions considered. The cell-centered, color function-based FPM with constant h_1 also out-performs the MDHF method for $N/D < 40$.

An important caveat when making these comparisons to the MDHF method is that we have defined the interface as $\mathcal{N}$ points that fall on the analytical interface; we have not reconstructed the interface from the VOF in a standard formalism. Any

interface reconstruction is expected to introduce additional error, particularly at lower resolutions. The error associated with reconstruction is considered in the next section.

2.4 Practical implementation with PLIC and G-grid

2.4.1 PLIC and G-grid description

Geometric reconstruction entails the approximation of interface shape and location in each interfacial control volume. The PLIC method is commonly used for this purpose [18, 104, 105]. In the PLIC method, the interface in each interfacial control volume is approximated by a plane. The plane is defined to have the same normal vector as the local VOF field, and the position of the plane is chosen so the fraction of the control volume under the plane (in Phase I) is equal to the VOF. The normal vector,

$$\mathbf{n} = \nabla\psi, \tag{2.30}$$

is computed from the first derivatives of the VOF, acquired through solution of Eqn. (2.18) with $h_1 = 2.8\Delta$. The PLIC algorithm implemented in this work is defined in detail by [119].

Advection of a poorly resolved geometric VOF results in unphysical distortion of the interface [104]. It is therefore beneficial to perform species transport on a G-grid in simulations of primary atomization, as demonstrated for level-set methods by [52] and VOF methods by [65]. A G-grid is a relatively fine computational grid that is coincident with the flow solver grid [48, 52]. The practical considerations for implementation are detailed by [48]. The degree of refinement of the G-grid is defined by the ratio of the flow solver grid spacing to the G-grid spacing, Δ/Δ_G. For example, a G-grid with three-times the resolution of the flow solver grid is defined by

$\Delta/\Delta_G = 3$. In this section, we consider the effects of using a G-grid on the accuracy of curvature computed with the FPM.

The PLIC algorithm is solved on the G-grid, resulting in a reconstructed interface in each interfacial G-grid cell. In a full fluid dynamics simulation, the VOF would also be transported on the G-grid. The VOF of a flow solver grid cell is computed as the aggregate of the G-grid VOF values that fall within its bounds. All curvature calculations are performed on the flow solver grid as a function of the flow solver VOF, but the interface location is defined by the PLIC surfaces on the G-grid. The computational cost of computing curvature is therefore independent of the level of G-grid refinement, Δ/Δ_G, because curvature is computed as a function of the VOF on the flow solver grid. The centroid of the PLIC surface in each interfacial G-grid cell serves as the interfacial point for curvature calculation – in contrast to the $\mathcal{N}$ points seeded on the interface in the previous section. In practice, one need only compute curvature at a single PLIC surface per flow solver grid cell. However, every PLIC surface is considered here to best quantify the errors.

2.4.2 Performance evaluation

The same numerical parameters and methodology used to evaluate curvature in Section 2.3 are used in this section. The VOF in each G-grid cell is initialized by dividing it into a $33 \times 33 \times 33$ sub-grid. If the interface falls within the bounds of a sub-grid cell, the VOF is initialized by assuming the interface is a plane with a normal vector corresponding to the $D = 0.4$ sphere. The VOF of the G-grid cell is then computed as the aggregate of the sub-grid VOF values.

Curvature errors for the cell-centered color function approach with $h_1^\star = 4$ and $h_2^\star = 3$ appear in Fig. 2.7. Four different resolutions of G-grid have been considered with PLIC interface reconstruction. The case of $\Delta/\Delta_G = 1$ uses a G-grid with the same resolution as the flow solver grid – this is equivalent to not using a G-grid and

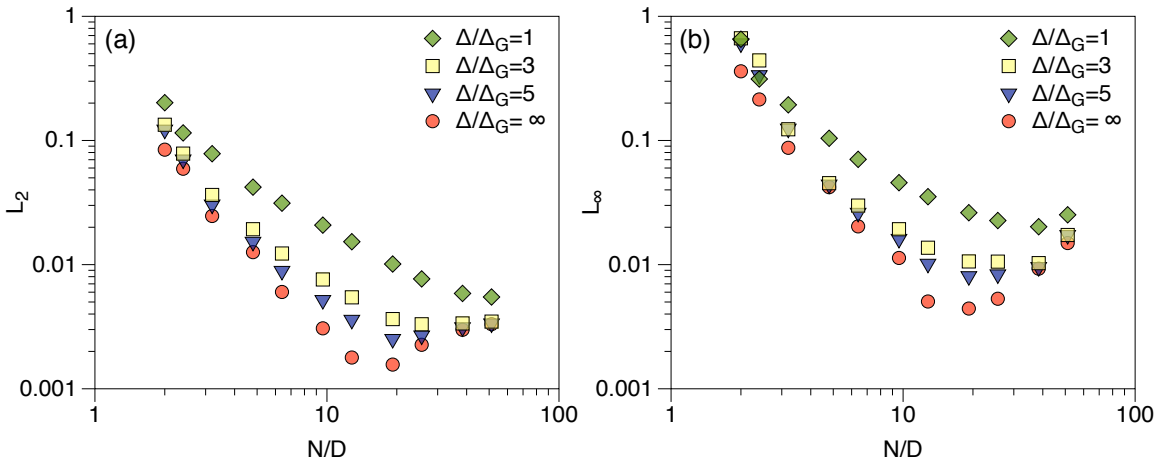

Figure 2.7. Comparison of errors produced with different levels of G-grid refinement for cell-centered calculations with $h_1^\star = 4\Delta$ and $h_2^\star = 3\Delta$ and PLIC interface reconstruction: (a) L_2 error norms; (b) L_∞ error norms.

simply using the flow solver grid. This case presents the largest error of the four considered. The case of $\Delta/\Delta_G = \infty$ uses a G-grid with infinite resolution, and is the equivalent of knowing the analytical position of the interface. This case presents the smallest error, and is the same result that appeared in Section 3 for $h_1^\star = 4$ and $h_2^\star = 3$. The final two cases, $\Delta/\Delta_G = 3$ and $\Delta/\Delta_G = 5$, dramatically improve accuracy relative to the case without a G-grid. The curvatures computed at the cell-centers are the same for all four cases in Fig. 2.7 because they are computed from the flow solver VOF. The differences in error are therefore entirely due to interpolation. It is apparent that the G-grid, in addition to improving phase transport behavior [48, 52], improves the accuracy of curvature in the context of convolved VOF methods by reducing interpolation error. In addition to reducing interpolation error, the PLIC surfaces on the G-grid also allow for interface-based calculations, as demonstrated by the hybrid FPM implementation in Fig. 2.8.

A hybrid FPM implementation that utilizes a G-grid and PLIC interfacial reconstruction, denoted by hybrid FPM #2, appears in Fig. 2.8. The details of the hybrid

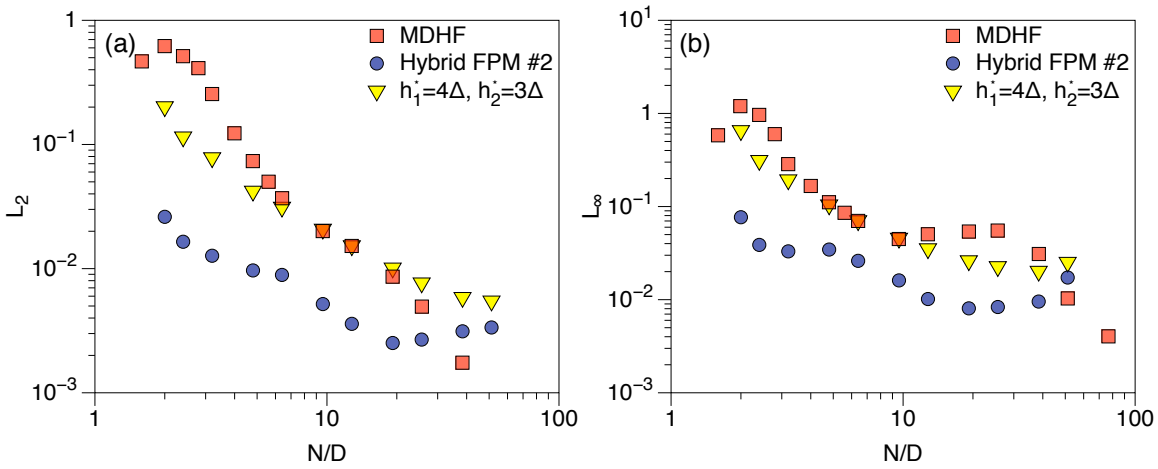

Figure 2.8. Performance comparison of hybrid FPM #2 (defined in Table 2.2), the color function-based FPM with cell-centered calculations and PLIC interface reconstruction (triangle), and the MDHF method: (a) L_2 error norms; (b) L_∞ error norms. MDHF results are taken from [105] .

implementation are defined in Table 2.2. Unlike hybrid FPM #1 from Section 2.3.7, hybrid FPM #2 uses a constant $h_1^\star = 4$. The $\Delta/\Delta_G = 1$ results from Fig. 2.7 and the MDHF results from [105] are also included for reference. Figure 2.8 demonstrates that using a standard PLIC reconstruction without a G-grid and selecting the influence radii to be $h_1^\star = 4$ and $h_2^\star = 3$ produces results comparable to the MDHF for $N/D < 50$. However, hybrid FPM #2 demonstrates superior accuracy for $N/D < 40$. Additional smoothing of the VOF could be introduced by increasing $h_1^\star$ as a function of N/D, but here we consider the case of constant $h_1^\star$, favoring computational affordability. The most prominent improvement of the hybrid implementation over the other two methods appears for $N/D < 25$, which is significant for spray simulation. It is possible to maintain high accuracy in this range while reducing computational cost.

Results for a third hybrid approach, hybrid FPM #3 (defined in Table 2.3), appear in Fig. 2.9. The figure also includes results for a VOF-based curvature with $h_2^\star = 3\Delta$. Both the VOF-based curvature and hybrid FPM #3 utilize a G-grid defined by

Table 2.2. Parameters for hybrid FPM #2 in Fig. 2.8.

Resolution	Interface/Cell-centered	$h_1^\star$	$h_2^\star$	Δ/Δ_G
$N/D < 5$	Interface	4Δ	3Δ	5
$N/D \geq 5$	Cell	4Δ	3Δ	5

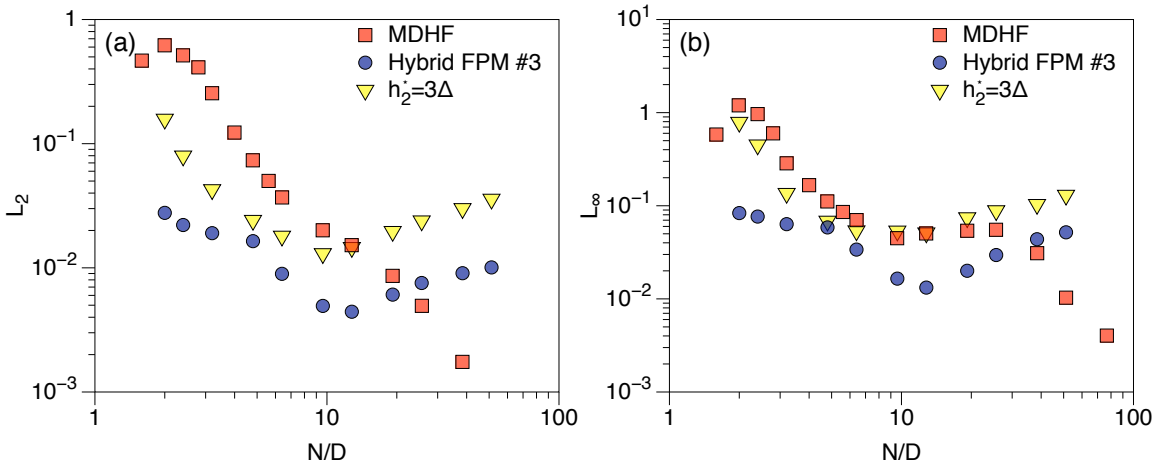

Figure 2.9. Performance comparison of hybrid FPM #3 (defined in Table 2.3), the VOF-based FPM with PLIC interface reconstruction and a G-grid of $\Delta/\Delta_G = 5$ (triangle), and the MDHF method: (a) L_2 error norms; (b) L_∞ error norms. MDHF results are taken from [105] .

$\Delta/\Delta_G = 5$. The effective influence radii used to compute curvatures in Fig. 2.9 are smaller than those used in Fig. 2.8. For example, Hybrid FPM #3 uses $h_1^\star = 2\Delta$, whereas hybrid FPM #2 uses $h_1^\star = 4\Delta$. The VOF-based curvature does not require a color function. Despite the reduction in influence radii, the curvatures in Fig. 2.9 remain relatively high quality for $N/D < 25$. Small influence radii, such as those used in Fig. 2.9, optimize accuracy and computational cost for features resolved by few computational nodes. The additional smoothing behavior of large influence radii, such as those used in Fig. 2.8, is required for flows with well-resolved features.

Table 2.3. Parameters for hybrid FPM #3 in Fig. 2.9.

Resolution	Interface/Cell-centered	$h_1^\star$	$h_2^\star$	Δ/Δ_G
$N/D < 5$	Interface	2Δ	3Δ	5
$N/D \geq 5$	Cell	2Δ	3Δ	5

2.5 Chapter summary and conclusions

We have presented two FPM formulations for computing curvature on poorly resolved, three-dimensional surfaces in the context of VOF methods. The first formulation is based on a convolved VOF, called the color function, and the second is based on the VOF directly. Both formulations have been implemented to compute the curvature of a sphere, and the results have been evaluated with respect to the L_2 and L_∞ error norms. We use the errors in calculated curvature to assess (*i.*) the relative quality of curvature computed at cell-centers and at interfaces, (*ii.*) the potential to reduce the computational cost of the color function-based calculations, (*iii.*) the relative quality of curvature computed from the color function and from the VOF, (*iv.*) the performance of a hybrid method with resolution-dependent parameters, and (*v.*) the impact of interface reconstruction and the use of a G-grid on accuracy.

Results show that curvatures computed at the cell center are more accurate than those computed at the interface for nearly all resolutions. Curvatures computed at the interface are less accurate because the interface moves relative to the mesh and causes differential weighting between nodes. For very low resolutions, interpolation from the cell center to the interface dominates the error. In this regime the interface-centered calculations are more accurate because they do not require interpolation. Regardless of where the calculations are performed, increasing the influence radii improves curvatures. Although no radii produce convergence, larger radii damp high frequency variations in the VOF, improving accuracy at higher resolutions. A hybrid

FPM approach leverages these concepts and demonstrates high accuracy across all tested resolutions.

Color function-based curvatures are up to ten times more accurate than VOF-based curvatures. The benefit of computing curvature from the VOF is the computational savings – it eliminates the convolution operation and reduces the expense of solving the FPM system by approximately 20%. Both the color function and VOF formulations out-perform the MDHF method for poorly resolved geometries. Significant performance improvements are achieved with the use of G-grids, which dramatically reduce interpolation error. A hybrid color function formulation, coupled with a G-grid and PLIC interfacial reconstruction, produces significantly smaller errors than the MDHF method for nearly the entire range of tested resolutions.

The resolutions considered in this work reflect those observed in contemporary spray simulations [51, 124, 22]. At these resolutions, the FPM demonstrates better accuracy than the height function method. The results suggest the FPM to be an improvement to the state-of-the-art for accurately computing curvature in VOF simulations of sprays.

Chapter 3

Dynamic evaluation of the finite particle method for computing curvature

This chapter has been adapted from Wenzel and Garrick [149].

3.1 Introduction

The length scale used for computing interfacial curvature is an adjustable parameter in the finite particle method defined in Chapter 2. The length scale is adjusted via the influence radius. The Adjustable Curvature Evaluation Scale (ACES) methodology presented by Owkes et al. [107] also has the functionality of a variable length scale. The important conclusions drawn by Owkes et al. [107] during the development of the ACES methodology are the length scale used to compute curvature significantly influences dynamic accuracy, and any evaluation of curvature methods should include dynamic testing. In this chapter, we evaluate the dynamic performance of the FPM

in simulations of poorly-resolved oscillating ethanol droplets in two and three dimensions. We explore the dependence of performance on the length scale considered when computing curvature, and we introduce the possibility of dynamically modifying the length scale. Performance comparisons are made with the second-order height function method, and a Coupled FPM/Height Function method is presented. We conclude the chapter with a summary of results and a discussion.

3.2 Methods

3.2.1 Governing equations

We consider the multiphase, incompressible Navier-Stokes equations for an immiscible system with phase-dependent properties. The system is governed by conservation of momentum,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla P + \frac{1}{\rho}\nabla \cdot \left[\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)\right], \tag{3.1}$$

where ρ is the density, $\mathbf{u}$ is the velocity vector, P is the pressure, and μ is the viscosity. The fluid is made incompressible by enforcing the divergence-free condition on the velocity field,

$$\nabla \cdot \mathbf{u} = 0. \tag{3.2}$$

Knowledge of which phase (liquid or gas) is located at $\mathbf{x} = \{x^1, x^2, x^3\}$ is required in order to compute the fluid properties. Evolution of the phase information is achieved by solving the Volume-of-Fluid (VOF) transport equation,

$$\frac{\partial \psi}{\partial t} + \mathbf{u} \cdot \nabla \psi = 0. \tag{3.3}$$

The VOF ψ is defined as the volume fraction of the liquid phase in a control volume, meaning $\psi = 1$ in cells filled with liquid, $\psi = 0$ in cells filled with gas, and $0 < \psi < 1$ in cells containing both phases. The viscosity and density are computed at all $\mathbf{x}$ as a

function of the VOF:

$$\mu(\mathbf{x}) = \mu_l \psi(\mathbf{x}) + \mu_g (1 - \psi(\mathbf{x})), \tag{3.4}$$

and

$$\rho(\mathbf{x}) = \rho_l \psi(\mathbf{x}) + \rho_g (1 - \psi(\mathbf{x})). \tag{3.5}$$

Interfacial dynamics enter the system through the boundary condition at the phase interface,

$$[P] = \sigma\kappa + 2[\mu]\mathbf{n}^T \cdot \nabla\mathbf{u} \cdot \mathbf{n}, \tag{3.6}$$

where $[P]$ is the pressure jump at the interface, σ is the surface tension coefficient, κ is the interfacial curvature, $[\mu]$ is the difference in fluid viscosities, and $\mathbf{n}$ is the interfacial unit normal vector. The surface tension $\sigma\kappa$ is directly proportional to the curvature κ. The Finite Particle Method computes the curvature κ as a function of the VOF ψ according to Eq. (2.29).

3.2.2 Description of flow solver

Our strategy for solving the governing equations is largely modeled off of the solver described by Desjardins et al. [21]. The momentum transport equation is solved on a staggered grid with a second-order spatial discretization. The VOF transport and momentum equations are staggered in time and are integrated in time with a semi-implicit Crank–Nicolson scheme (in an explicit iterative configuration). VOF transport and interfacial reconstruction is accomplished with the geometric approach outlined in [152], and the algebraic PLIC reconstruction methods of [119]. The pressure jump at the interface due to surface tension is imposed with the Ghost Fluid Method (GFM) [32]. Trilinear interpolation is used to determine the curvature at the centroid of the reconstructed interface in the interfacial cells (where $0 < \psi < 1$), and this curvature defines the jump condition for pressure in the GFM. Variable viscosity is not treated with the GFM, but rather treated as in [21].

The weight function used to compute the color function is defined by

$$W_1(R, h_1) = \begin{cases} \left(1 - R^2\right)^6, & 0 \leq R \leq 1, \\ 0, & R > 1, \end{cases} \tag{3.7}$$

where $R = |\mathbf{x}_j - \mathbf{x}_o| / h_1$. The weight function used to construct the FPM sampling basis Ω is defined by

$$W_2(R, h_2) = \begin{cases} \left(-3R^4 + 8R^3 - 6R^2 + 1\right), & 0 \leq R \leq 1, \\ 0, & R > 1, \end{cases} \tag{3.8}$$

where $R = |\mathbf{x}_j - \mathbf{x}_o| / h_2$ [87]. The influence radii h_1 and h_2 are free parameters. For all simulations presented in this chapter, the color function influence radius is fixed to $h_1 = 1.5\Delta$. The FPM influence radius h_2 is varied to determine its impact on dynamic performance.

3.3 Performance assessment

Oscillating ethanol droplets suspended in air are considered as a dynamic test platform for the FPM. Test conditions are limited to low resolutions to reflect the degree of resolution expected for droplets and ligaments present in contemporary spray simulations. The evaluation is divided into three parts: oscillations of two-dimensional droplets as a function of the mesh resolution and the FPM influence radius h_2; oscillations of three-dimensional droplets with comparison between the FPM and a second-order height function method; and oscillations of two-dimensional droplets with adaptive influence radii.

3.3.1 Oscillating two-dimensional ethanol droplets

Problem setup

An ethanol droplet with radius $a = 20$ μm is suspended in air. The droplet interface is initially perturbed into an elliptical shape defined by

$$\frac{x^2}{(a/0.75)^2} + \frac{y^2}{(0.75a)^2} = 1, \tag{3.9}$$

where the origin is centered on the droplet. The domain is a periodic square with side length $L = 100$ μm. The ethanol has a density of $\rho_l = 789.3$ kg/m^3 and a viscosity of $\mu_l = 1.2 \times 10^{-3}$ Pa $\cdot$ s, and the air has a density of $\rho_g = 1.225$ kg/m^3 and a viscosity of $\mu_g = 1.81 \times 10^{-5}$ Pa $\cdot$ s. The surface tension coefficient is $\sigma = 0.02178$ N/m [143]. Mesh resolutions of $N_x = 8$, $N_x = 16$, $N_x = 32$, and $N_x = 64$ are considered. The FPM influence radius is considered at two values, $h_2 = 2.25\Delta$ and $h_2 = 4\Delta$.

Potential energy stored in the elliptical interface produces temporal shape oscillations of the droplet. The kinetic energy provides a natural means to examine the oscillation behavior, as the problem is characterized by an exchange between kinetic and potential energy. This work considers the dimensionless droplet kinetic energy $E^\star$,

$$E^\star = \frac{1}{E_o} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2} \rho_l \psi \mathbf{u} \cdot \mathbf{u} dx dy, \tag{3.10}$$

where E_o is a characteristic energy (defined on a case-by-case basis). Time is made dimensionless according to Owkes et al. [107]

$$t^\star = t \left(2\pi \sqrt{\frac{(\rho_l + \rho_g)\, a^3}{6\sigma}} \right)^{-1}. \tag{3.11}$$

The expected behavior of $E^\star$ is a smooth oscillation between $E^\star \approx 0$ and $E^\star \approx 1$ with approximately constant frequency and decreasing amplitude with time.

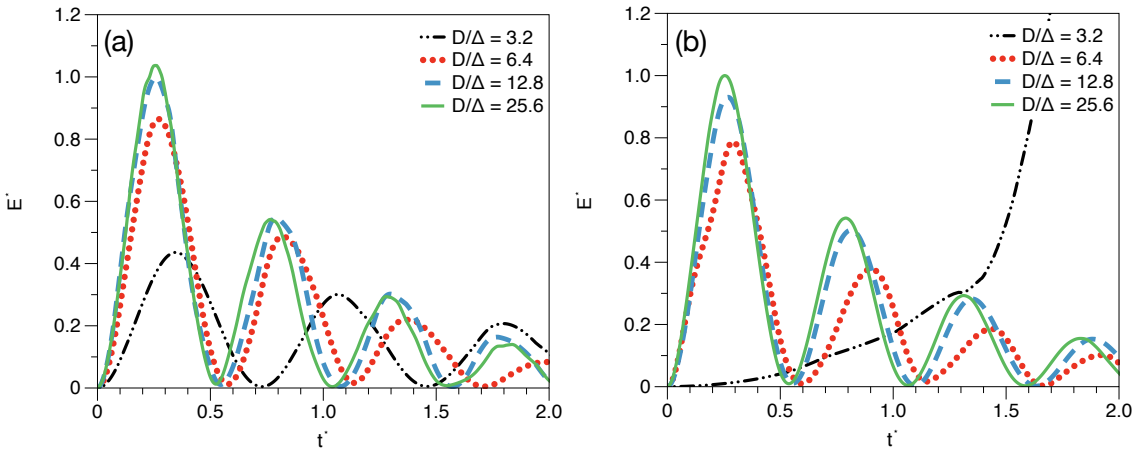

Figure 3.1. Dimensionless kinetic energy of oscillating 2D droplets for two weight function configurations: (a) $h_1 = 1.5\Delta$ and $h_2 = 2.25\Delta$; (b) $h_1 = 1.5\Delta$ and $h_2 = 4\Delta$.

Results

Figure 3.1 presents the dimensionless kinetic energy for the two-dimensional droplet simulations. Oscillations presented in Fig. 3.1 panel *a* are computed with $h_2 = 2.25\Delta$ and oscillations presented in Fig. 3.1 panel *b* are computed with $h_2 = 4\Delta$. Panels *a* and *b* each include simulation results performed on four different meshes with resolutions ranging from $N_x = 8$ through $N_x = 64$. The simulations are identified by the number of grid points spanning the undeformed droplet diameter D/Δ in order to clearly express the extent to which the droplet is resolved. In both panels *a* and *b*, the characteristic kinetic energy E_o is selected as the maximum kinetic energy achieved in the simulation with $h_2 = 4\Delta$ and $D/\Delta = 25.6$.

The narrow influence radius of $h_2 = 2.25\Delta$, presented in Fig. 3.1 panel *a*, performs well for $D/\Delta = 3.2$ and $D/\Delta = 6.4$, considering the general oscillatory dynamic is achieved despite dramatic under-resolution. However, convergence is not observed as the resolution increases. The highest resolution case, $D/\Delta = 25.6$, presents high-frequency anomalies that imply the curvature is poorly characterized. The Fig. 3.1 panel *a* results show that the narrow influence radius of $h_2 = 2.25\Delta$ performs well for

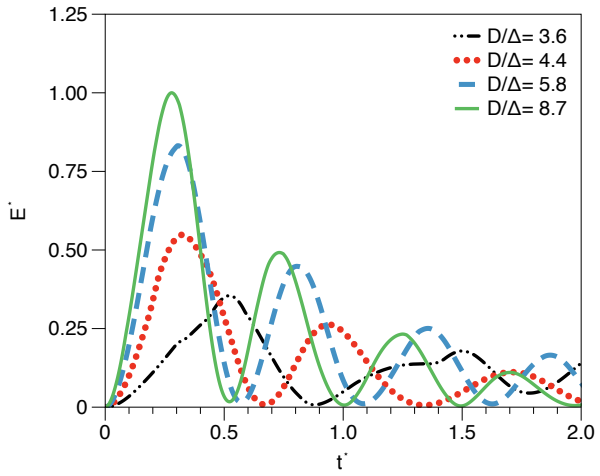

Figure 3.2. Dimensionless kinetic energy of oscillating 3D droplets with $h_1 = 1.5\Delta$ and $h_2 = 2.3\Delta$.

poorly-resolved droplets, such as $D/\Delta = 3.2$, but poorly for well-resolved droplets, such as $D/\Delta = 25.6$. The opposite trend appears in Fig. 3.1 panel *b* for $h_2 = 4\Delta$. When the influence radius is large, as in the $h_2 = 4\Delta$ simulations presented in Fig. 3.1 panel *b*, the oscillations of well-resolved droplets are smooth and regular. However, poorly resolved droplets, notably $D/\Delta = 3.2$, fail catastrophically because the large influence radius provides too much smoothing. In this case, the interfacial dynamics become unphysical. The results presented in Fig. 3.1 are consistent with the findings of Owkes et al. [107]: the length scale used to compute curvature (the influence radius) should correspond to the length scales of perturbations on the interface. If the length scale used to compute curvature is too large or too small with respect to interfacial perturbations, performance suffers.

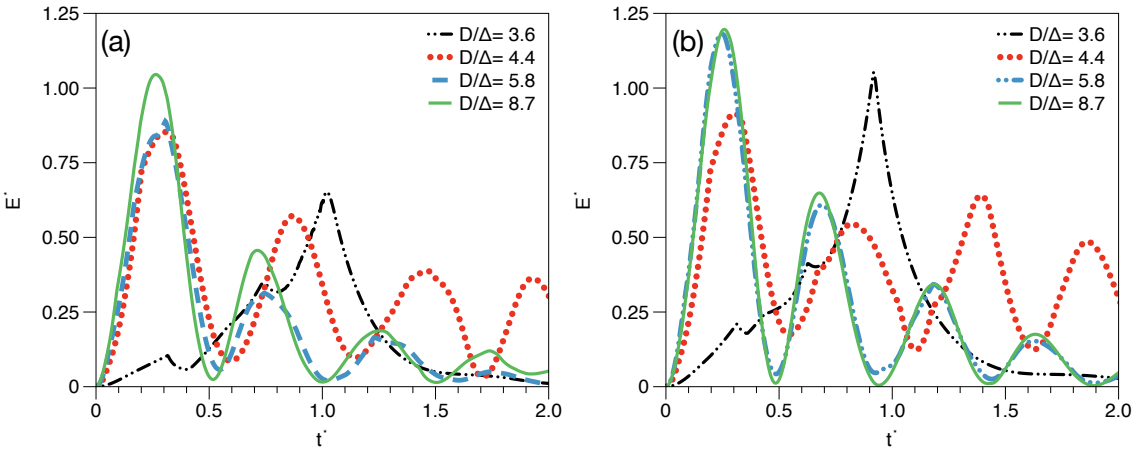

Figure 3.3. Dimensionless kinetic energy of oscillating 3D droplets computed with a second-order height function method: (a) column heights of 3Δ; (b) column heights of 5Δ.

3.3.2 Oscillating three-dimensional ethanol droplets

Problem setup

Three-dimensional calculations are performed in generally the same way as the two-dimensional cases. A droplet is placed in a periodic cube with side length $L = 100$ μm, and the interface is perturbed according to

$$\frac{x^2}{(a/0.75)^2} + \frac{y^2}{(0.75a)^2} + \frac{z^2}{(0.75a)^2} = 1, \tag{3.12}$$

where $a = 20$ μm. Mesh resolutions of $N_x = 10$, $N_x = 12$, $N_x = 16$, and $N_x = 24$ are considered.

Instead of examining the significance of influence radii, we directly compare one implementation of the FPM with $h_1 = 1.5\Delta$ and $h_2 = 2.3\Delta$ to the height function method. The height function method demonstrates second-order convergence at high resolutions, and is well-documented in the literature [97, 18, 105]. The implementation of the height function method considered here is second-order accurate and uses column heights of either 3Δ or 5Δ.

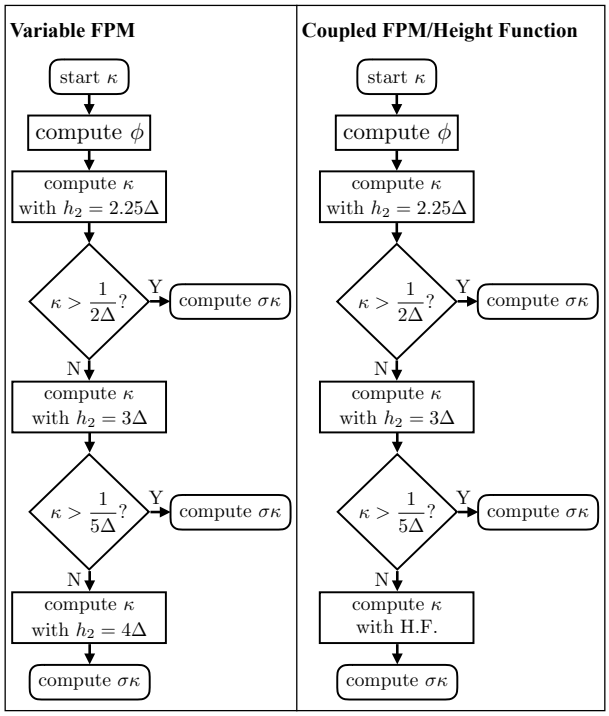

Figure 3.4. Logic flow charts for the Variable FPM method and the Coupled FPM/Height Function method. Both methods are implemented in Fig. 3.5.

Results

Figure 3.2 presents the dimensionless kinetic energies of the three-dimensional droplets computed with the FPM using $h_1 = 1.5\Delta$ and $h_2 = 2.3\Delta$. The characteristic energy E_o is taken to be the maximum kinetic energy achieved by the $D/\Delta = 8.7$ simulation. The FPM generally presents the correct oscillatory behavior for all resolutions considered. At the lowest resolution of $D/\Delta = 3.6$, the amplitude is low and the period is irregular, but the droplet still oscillates. At all higher resolutions, the oscillations are smooth and well-behaved. The height function results are presented in Fig. 3.3. The results in Fig. 3.3 panel *a* are produced with column heights of 3Δ and the results in Fig. 3.3 panel *b* are produced with column heights of 5Δ. Neither height function method presents oscillatory behavior at the lowest resolution of $D/\Delta = 3.6$. The shorter height function implementation in Fig. 3.3 panel *a* does not present smooth

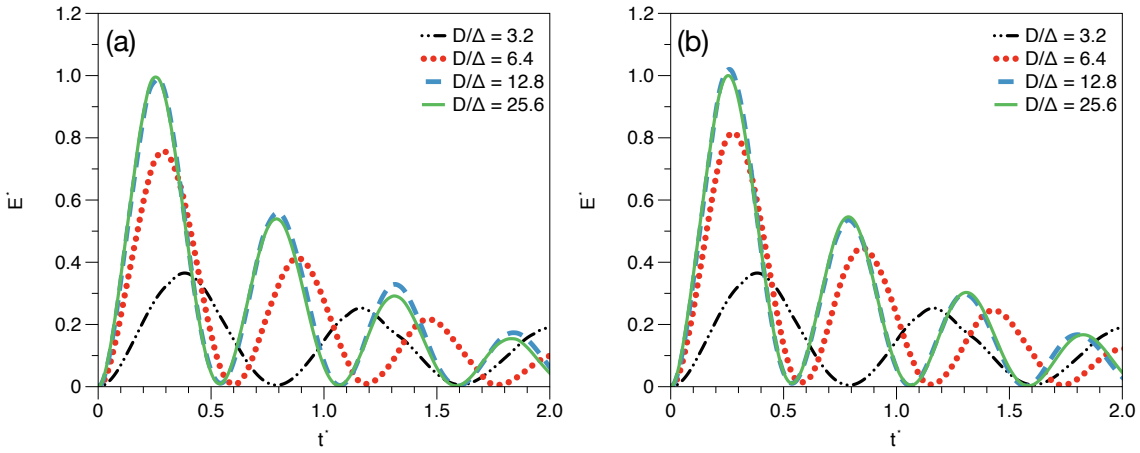

Figure 3.5. Dimensionless kinetic energy of oscillating 2D droplets computed with (a) the Variable FPM and (b) the Coupled FPM/Height Function.

oscillations for any tested resolution, and the taller height function implementation in Fig. 3.3 panel *b* does not present smooth oscillations until $D/\Delta = 8.7$. The amplitude of the height function oscillations with column heights of 5Δ is greater than the amplitude of the FPM oscillations. At the highest resolution of $D/\Delta = 8.7$, the oscillations produced with the height function method with column heights of 5Δ are smooth, and this configuration is expected to perform well at all higher resolutions.

3.3.3 Adaptive methods

The results presented in the previous sections show that dynamic performance of the FPM depends strongly on the influence radius, in agreement with the findings of [107]. It is therefore desirable to adaptively modify the influence radius such that it is appropriate for the local length scales. A basic adaptive influence radius FPM can be implemented by first computing curvature with a small influence radius, followed by a check for low curvature regions (these are the regions that are well-resolved), followed by a re-calculation of curvature in well-resolved regions with a larger influence radius. Figure 3.4 presents logic diagrams for two adaptive FPM methods. The first, called

the Variable FPM, uses three different influence radii h_2 as a function of the local curvature κ. The second, called the Coupled FPM/Height Function, uses the FPM for high curvature regions and a second-order height function with column heights of 5Δ for low curvature (well-resolved) regions.

Figure 3.5 presents the dynamic performance of the Variable FPM (panel *a*) and the Coupled FPM/Height Function method (panel *b*) for the two-dimensional droplet configuration. The characteristic energy E_o is taken to be the maximum kinetic energy achieved by the Coupled FPM/Height Function with $D/\Delta = 25.6$. Both the Variable FPM and the Coupled FPM/Height Function demonstrate smooth and regular oscillatory behavior at high and low resolutions. The performance of both methods is similar, but not identical. The results in Fig. 3.5 suggest that the adaptive FPM approaches described in Fig. 3.4 out-perform the fixed influence radius FPM and the Height Function method. Improved performance is attributable to monitoring and dynamically enforcing compatibility of the influence radius with the interfacial length scales.

3.4 Chapter summary and conclusions

We have simulated two and three-dimensional oscillating ethanol droplets with the Finite Particle Method (FPM) and Height Function (HF) in order to evaluate dynamic performance. The numerical resolutions of the simulations reflect the resolutions of droplets and ligaments present in contemporary spray simulations. Low resolution, three dimensional droplets simulated with the FPM present smooth, regular oscillations, whereas droplets simulated with the Height Function do not oscillate smoothly until a minimum degree of resolution is achieved. Two-dimensional results show that good performance requires the length scale considered by the FPM calculation (the FPM influence radius) to be comparable with the length scales of the interface: when the length scales of the interface are large, the influence radius of the FPM must be

large, and when the length scales of the interface are small, the influence radius of the FPM must be small. When there is compatibility between length scales, the FPM performs well at all resolutions considered in this work. Two methods – the Variable FPM and the Coupled FPM/Height Function – have been presented that adaptively modify the influence radius to satisfy length scale compatibility. Both adaptive methods perform well for all resolutions considered. Because the FPM can accurately compute curvature at all numerical resolutions present in spray simulations, it is expected to out-perform the height function method with respect to characterizing interfacial dynamics including capillary breakup. For this reason, the FPM is a valuable new tool for use in the numerical simulation of sprays.

Chapter 4

A point mass particle method for the simulation of immiscible multiphase flows on an Eulerian grid

4.1 Introduction

Many different methods have been developed and successfully implemented for determining the location of each phase, most prominently front-capturing [58, 103, 131, 130, 100, 92, 101, 21, 48], including modern implementations of the level set method [20, 48, 27, 28, 78, 5] and the volume of fluid (VOF) method [47, 152, 13, 104, 106]. Front-tracking [95, 139, 136, 37, 134], particle-based [45, 44, 82, 26], and fully Lagrangian methods [96, 98, 83, 125, 75, 133] are also widely used. Each of these

techniques have shortcomings which may include reinitialization, regularization, conservation error, prohibitive compute times, inaccurate description of surface tension, restrictions on the number of fluids considered, or numerical diffusion near interfaces [136, 27, 40, 48, 124]. Performance improvements are still required for simulation tools used for discovery (direct numerical simulation) and for practical purposes (large eddy simulation). A common strategy among both historical and contemporary methodologies is to couple Lagrangian and Eulerian frameworks.

Methods that couple Lagrangian particles with an Eulerian mesh have been proposed in a variety of contexts. Lagrangian surface meshes have been used to define implicit surface tension forces in Eulerian simulations [120], and have also been used to assist in solution of the Poisson equation on hybrid Eulerian-Lagrangian meshes [158]. Particle disorganization in Lagrangian methods adversely affects the fluid pressure computed with Eulerian kernels, and the use of an Eulerian background mesh has been shown to enhance the quality of pressure computed in moving particle semi-implicit methods [145]. (Wang et al. [145] also provides a thorough review of other efforts by researchers to use Eulerian grids, arbitrary Lagrangian-Eulerian methods, and re-mesh methods to overcome the numerical challenges associated with poorly-organized Lagrangian particles.) Interfacial regions have been treated with fully-Lagrangian smooth particle hydrodynamics, while interior regions retain an Eulerian treatment [90]. Particle-in-cell (PIC) methods use particles to determine convective transport, and a fixed grid to determine other dynamics including diffusion [44, 26]. Particle level set values provide corrections to the Eulerian level set in semi-Lagrangian level set methods [27, 28, 78, 5]. Lagrangian level set methods do not directly evolve an Eulerian level set, but instead communicate a Lagrangian level set to the Eulerian mesh [56]. Aspects of the Lagrangian level set [56] and the marker and cell method [45] have been coupled with the continuum body force method [96] to compute surface tension on particles, which is then communicated to the Eulerian

mesh in a method called the Lagrangian volume of fluid (LVOF) [85, 150, 146, 147].

In the Lagrangian level set, LVOF, and marker and cell methods, the phases are tracked with massless or infinitesimal fluid particles, also referred to as tracer particles. Updating the particle position requires communicating the Eulerian velocity to the particles through interpolation or some other method of weighting the Eulerian velocities at the particle location. Unfortunately interpolation of a divergence-free velocity does not produce a divergence-free particle velocity field [13]. Schemes exist to enforce the divergence-free condition on the particle field in the context of PIC simulations [144], but this approach does not generally satisfy the needs of an immiscible system. Merging and diverging characteristics have the propensity to create particle-dense and particle-sparse regions because particles follow the flow characteristics [27]. This property is one reason why many particle-based methods require particles to be added and deleted periodically, or for particles to be regularized every time-step [27, 120, 158, 26, 56]. Without regularization or reseeding, an initially well-distributed set of particles becomes poorly distributed over time, impacting both accuracy and Eulerian conservation. Additionally, if the particle field has the ability to resolve more Fourier modes than the Eulerian mesh (if the number of particles is greater than the number of Eulerian grid points), aliasing errors can arise in the harmonics of the principle Eulerian modes – this is the origin of the ringing instability in PIC methods [10, 11]. Strong velocity gradients near the interface of two phases – caused by variations in fluid properties and/or surface tension – can exacerbate the errors. Liu et al. [82] enforce a spatially uniform number density on a set of particles seeded in one of two phases by solving an elliptic Poisson equation. They fix the location of surface particles, preserving surface errors during particle relaxation, so it is unclear if the method conserves volume in practical flows.

Challenges with particle organization are not limited to coupled Eulerian-Lagrangian

methods. A detailed review of modern developments in Lagrangian simulation techniques is provided by Gotoh and Khayyer [42], including an exploration of the current state of accuracy, stability, and conservation properties. Of particular relevance to the present work is a discussion of particle regularization schemes used in projection-based, incompressible Lagrangian methods (either moving particle semi-implicit methods (MPS) [69] or incompressible SPH). Particle regularization schemes are used to keep particles well-distributed, and two commonly used methods are Particle Shifting (PS) [156, 79] and Dynamic Stabilization (DS) [137]. The PS method uses Fickian diffusion to keep particles well-distributed, while the DS method uses momentum-conserving, inter-particle repulsive forces. The relative merits of these methods are outlined by Khayyer et al. [67], where an optimized PS (OPS) scheme is proposed. These particle correction schemes for Lagrangian methods remain an active area of research [68].

This chapter presents a new framework for the simulation of immiscible multiphase flows. The Navier-Stokes equations are discretized on an Eulerian mesh, and the phase information is tracked by a set of particles corresponding to each phase. The particles are not tracer particles, but point mass particles (PMPs) defined by a position in space, a finite amount of mass, and a phase indicator. Associating the particles with a finite amount of mass restricts their spatial organization, such that the distribution of PMPs in space must reflect the fluid density. We derive an expression that relates the PMP velocity to the interpolated Eulerian velocity and propose a corresponding numerical model. The model (1) does not impose additional time-step or stability requirements, (2) allows for multi-scale resolution of the interface, and (3) allows for the simulation of more than two fluids. We perform phase-tracking tests to assess performance of the method with respect to shape error and conservation error, and conclude with a series of dynamic test cases, including expanding concentric ellipses, stationary and oscillating droplets, a droplet in shear flow, and a Rayleigh-Taylor

instability.

4.2 Formulation

4.2.1 Point mass particle discretization

Consider a fluid system comprised of N_ζ phases ζ^α, identified by $\zeta^1, \zeta^2, ..., \zeta^{N_\zeta - 1}, \zeta^{N_\zeta}$. The spatial distribution of phase is defined by a phase indicator function χ^α corresponding to each phase ζ^α. The phase indicator corresponding to phase ζ^α is a binary function given by

$$\chi^\alpha(t, \mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \zeta^\alpha, \\ 0, & \mathbf{x} \notin \zeta^\alpha, \end{cases} \tag{4.1}$$

where $\mathbf{x} = \{x, y, z\}$. In the special case of two-phase flow, the spatial distribution of both phases is defined by a single phase indicator function χ,

$$\chi(t, \mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \zeta^1, \\ 0, & \mathbf{x} \in \zeta^2. \end{cases} \tag{4.2}$$

The subdomain $\mathbf{x} \in \zeta^\alpha$ is discretized by N_α Lagrangian particles. Each particle i is associated with a mass, such that the total mass of phase ζ^α is distributed amongst the N_α particles, satisfying

$$\int_V \rho^\alpha \chi^\alpha dV = \sum_{i=1}^{N_\alpha} M_i, \tag{4.3}$$

where ρ^α is the density of phase ζ^α, M_i is the mass of particle i, and V is the domain volume (summation is not implied by repeated superscripts in this expression). In addition to mass M_i, each particle is associated with a position $\mathbf{X}_i$ (capital to emphasize the position as a property of particle i) and a Lagrangian phase indicator for each phase: particles belonging to phase ζ^α are identified by $\chi_i^\alpha = 1$ and particles

belonging to a different phase are identified by $\chi_i^\alpha = 0$. (In the context of two-phase flow, similar to Eq. (4.2), particles belonging to phase ζ^1 are assigned $\chi_i = 1$ and particles belonging to phase ζ^2 are assigned $\chi_i = 0$.) The PMPs are therefore a discretization of the mass and the associated phase in the domain.

The fluid density at point $\mathbf{x}$ is approximated by convolving the mass M_i of every particle near $\mathbf{x}$ with a compact weight function W according to

$$\rho(\mathbf{x}) = \sum_{i=1}^{N_p} W\left(\mathbf{x} - \mathbf{X}_i, h\right) M_i, \tag{4.4}$$

where N_p is the total number of particles belonging to any phase ζ^α within the compact support radius h of position $\mathbf{x}$ [83, 113]. The weight function $W\left(\mathbf{x} - \mathbf{X}_i, h\right)$ is defined as compact because $W\left(\mathbf{x} - \mathbf{X}_i, h\right) = 0$ for $|\mathbf{x} - \mathbf{X}_j| \geq h$. N_p appearing in Eq. (4.4) is therefore the total number of particles of any phase ζ^α whose position $\mathbf{X}_i$ satisfies $|\mathbf{x} - \mathbf{X}_i| < h$, where h is the "compact support radius" of the weight function W. In other words, N_p is the number of particles who make a non-zero contribution to the summation in Eq. (4.4); the position $\mathbf{X}_i$ of each of these particles satisfies $|\mathbf{x} - \mathbf{X}_i| < h$.

Particles move with velocity $\mathbf{U}_i$ and accelerate due to pressure gradients and surface tension according to

$$\frac{d\mathbf{U}_i}{dt} = \mathbf{P}_i + \mathbf{F}_i, \tag{4.5}$$

where $\mathbf{P}_i$ is the acceleration vector due to pressure and $\mathbf{F}_i$ is the acceleration vector due to surface tension. The specific form of $\mathbf{P}_i$ considered here is widely used to compute pressure acceleration in SPH simulations [93], and is derived from Eq. (4.4), an equation of state for pressure $P_i = f\left(\rho_i\right)$, and the principle of least action:

$$\mathbf{P}_i = -\sum_{j=1}^{N_p} \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2}\right) M_j \nabla W_{ij}, \tag{4.6}$$

where $\nabla W_{ij} = \nabla W\left(\mathbf{X}_i - \mathbf{X}_j, h\right)$ is the gradient of the weight function W evaluated at $\mathbf{X}_j$ relative to $\mathbf{X}_i$ [83, 113].

Immiscible multiphase systems include surface tension arising from molecular-scale interactions at the interface, resulting in acceleration $\mathbf{F}_i$. A variety of models are available to define $\mathbf{F}_i$ and specific forms are specified later [96, 133].

4.2.2 Coupling of the point mass particle velocity and Eulerian velocity

We consider a multiphase system where the fluid velocity $\mathbf{u}$ is known at a set of grid points and the phase information and mass has been discretized by a set of PMPs. The particles move with velocity $\mathbf{U}_i$ (defined by Eq. (4.5)) over a small time increment dt according to

$$d\mathbf{X}_i = \mathbf{U}_i dt. \tag{4.7}$$

An approximation of the velocity $\mathbf{U}_i$ of particle i located at $\mathbf{X}_i$ can be made by tri-linearly interpolating the fluid velocity $\mathbf{u}$ from the surrounding grid points to the location $\mathbf{x} = \mathbf{X}_i$. We denote the interpolated velocity by $\overline{\mathbf{u}}_i = \mathbf{G}(\mathbf{u}, \mathbf{X}_i)$, where $\mathbf{G}$ is the interpolation operator. The velocity $\overline{\mathbf{u}}_i$ approximates the velocity of an infinitesimal fluid particle or a tracer particle, rather than a PMP. Only in the limit as the particle mass goes to zero does a point mass particle behave as an infinitesimal fluid particle:

$$\lim_{M_i \to 0} \mathbf{U}_i \to \mathbf{u}(\mathbf{X}_i). \tag{4.8}$$

We decompose $\mathbf{U}_i$ into the interpolated velocity $\overline{\mathbf{u}}_i$ and a residual velocity $\mathbf{u}'_i$:

$$\mathbf{U}_i = \overline{\mathbf{u}}_i + \mathbf{u}'_i. \tag{4.9}$$

The interpolated velocity $\overline{\mathbf{u}}_i$ describes the velocity field resolved on the Eulerian mesh interpolated to the particle location. It includes errors associated with the interpolation scheme. The residual velocity, $\mathbf{u}'_i$, is comprised of the negative of the errors introduced by interpolation, and the motions required to satisfy the particle acceleration equation, Eq. (4.5). Assuming the interpolated velocity and the point mass

velocities are smooth functions in time, and therefore differentiable, we substitute the decomposed velocity into the acceleration equation for the PMP, Eq. (4.5), to yield

$$\frac{d\mathbf{U}_i}{dt} = \frac{d\overline{\mathbf{u}}_i}{dt} + \frac{d\mathbf{u}'_i}{dt} = \mathbf{F}_i + \mathbf{P}_i. \tag{4.10}$$

Further, we assume residual acceleration terms for pressure $\mathbf{P}'_i$ and surface tension $\mathbf{F}'_i$ exist such that

$$\frac{d\mathbf{u}'_i}{dt} = \mathbf{F}'_i + \mathbf{P}'_i \tag{4.11}$$

and

$$\frac{d\mathbf{U}_i}{dt} = \frac{d\overline{\mathbf{u}}_i}{dt} + \mathbf{F}'_i + \mathbf{P}'_i. \tag{4.12}$$

Having decomposed the particle acceleration into Eulerian and Lagrangian terms, we proceed to develop an expression for velocity $\mathbf{U}_i$ as a function of time. Performing a backward Taylor series expansion between time levels n and $n-1$ separated by Δt yields

$$\mathbf{U}_i^{n-1} = \mathbf{U}_i^n - \frac{d\mathbf{U}_i}{dt}^n \Delta t - \mathcal{O}\left(\Delta t^2\right), \tag{4.13}$$

or

$$\mathbf{U}_i^n = \mathbf{U}_i^{n-1} + \frac{d\mathbf{U}_i}{dt}^n \Delta t + \mathcal{O}\left(\Delta t^2\right). \tag{4.14}$$

Substituting the velocity and acceleration decompositions into the right-hand-side relates the point mass velocity $\mathbf{U}_i$ to the interpolated and residual velocities,

$$\mathbf{U}_i^n = \overline{\mathbf{u}}_i^{n-1} + \frac{d\overline{\mathbf{u}}_i}{dt}^n \Delta t + \mathbf{u}_i'^{n-1} + \frac{d\mathbf{u}'_i}{dt}^n \Delta t + \mathcal{O}\left(\Delta t^2\right). \tag{4.15}$$

After collapsing all time derivatives of the interpolated velocity and substituting Eq. (4.11), we arrive at an expression for the point mass particle velocity as a function of the interpolated velocity plus residual acceleration and residual velocity terms

$$\mathbf{U}_i^n = \overline{\mathbf{u}}_i^n + \left(\mathbf{F}'_i + \mathbf{P}'_i\right)^n \Delta t + \mathbf{u}_i'^{n-1} + \mathcal{O}\left(\Delta t^2\right). \tag{4.16}$$

Given an Eulerian velocity field, the interpolated velocity term $\overline{\mathbf{u}}_i$ is known. The residual acceleration terms for pressure $\mathbf{P}'_i$ and surface tension $\mathbf{F}'_i$, and the residual velocity $\mathbf{u}'_i$ require closure models.

4.2.3 Closure strategy

In this work we assume the fluid to be incompressible and the Eulerian velocity field to be divergence-free. Each particle is then assumed to have density ρ_1 and mass M_1. Every particle must therefore have the same residual pressure P_1' defined by the residual equation of state, $P_1' = f'(\rho_1)$. The modeled residual acceleration due to pressure is then a function only of the particle positions and the pressure weight function W^a,

$$\mathbf{P}_i' = -\sum_{j=1}^{N_p} \left(\frac{P_1'}{\rho_1^2} + \frac{P_1'}{\rho_1^2} \right) M_1 \nabla W_{ij}^a = -2\frac{P_1' M_1}{\rho_1^2} \sum_{j=1}^{N_p} \nabla W_{ij}^a = -c_1 \sum_{j=1}^{N_p} \nabla W_{ij}^a. \quad (4.17)$$

The parameter c_1 determines the extent of particle response to variations in particle number density and particle organization, and therefore determines the stability characteristics of the particle field. In the case of a compressible fluid, the assumptions made to derive Eq. (4.17) would be inapplicable. Instead, every particle would have a unique density, and a residual equation of state $f'(\rho)$ would need to be derived. The stability of the method would depend on the equation of state $f'(\rho)$. Further discussion of the compressible flow case is left for future work. The model parameters c_1 and ∇W^a in Eq. (4.17) will be specified in Section 4.3.2.

We make use of the pairwise force method to close the residual acceleration due to surface tension [132, 133]. The approach is well–validated in the context of SPH. The residual acceleration is given by

$$\mathbf{F}_i' = \sum_{j=1}^{N_p} -F_{\chi_i^\alpha, \chi_j^\alpha}^{int} \left(|\mathbf{r}_{ij}| \right) \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|}, \quad (4.18)$$

where $\mathbf{r}_{ij} = \mathbf{X}_i - \mathbf{X}_j$, and the subscript $\chi_i^\alpha, \chi_j^\alpha$ denotes that the inter-particle function $F_{\chi_i^\alpha, \chi_j^\alpha}^{int}$ depends on the phase of particles i and j. The function $F_{\chi_i^\alpha, \chi_j^\alpha}^{int}$ can be defined to produce the surface tension coefficient corresponding to a specific multiphase system [133]. In the present work, $\mathbf{F}_i'$ serves simply to reduce mixing of the phases at scales

below the resolution of the Eulerian mesh, and it is therefore not required to match the surface tension coefficient of the multiphase system. Surface tension dynamics relevant to the fluid system are considered in the Navier-Stokes equations (to be discussed later), rather than in Eq. (4.18). We generalize $F^{int}_{\chi_i^\alpha,\chi_j^\alpha}$ by defining

$$F^{int}_{\chi_i^\alpha,\chi_j^\alpha} = c_2 W^b + c_3 W^c. \tag{4.19}$$

where W^b is the weight function corresponding to the interaction between particles of the same phase ($\chi_i^\alpha = \chi_j^\alpha$), W^c is the weight corresponding to particles of different phases ($\chi_i^\alpha \neq \chi_j^\alpha$), and c_2 and c_3 are scaling constants. Each of these model parameters will be specified in Section 4.3.2.

Combining the models for residual accelerations due to pressure and surface tension returns a model for the point mass particle velocity,

$$\mathbf{U}_i^n = \overline{\mathbf{u}}_i^n - \sum_{j=1}^{N_p} \left[c_1 \nabla W_{ij}^a + \left(c_2 W_{ij}^b + c_3 W_{ij}^c \right) \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \right]^n \Delta t + \mathbf{u}_i'^{n-1}. \tag{4.20}$$

4.3 Numerical method

4.3.1 Eulerian-Lagrangian communication

The Eulerian grid requires phase information from the particles so that fluid properties and interfacial geometry can be computed. We use a color function (CF) approach to ensure smooth interfaces with perfect boundedness. The CF is used in the same way as the volume fraction for determining fluid properties. For two-phase flows, the color function is computed via convolution of the simplified two-phase Lagrangian phase indicator values χ_j, where $\chi_j = 1$ if particle j belongs to phase ζ^1 and $\chi_j = 0$ if particle j belongs to phase ζ^2. The convolution is performed at each Eulerian cell-center $\mathbf{x}_i$ according to

$$\phi_i = \frac{\sum_{j=1}^{N_p} \chi_j W_{ij}^\phi \left(\mathbf{x}_i - \mathbf{X}_j, h^\phi \right)}{\sum_{j=1}^{N_p} W_{ij}^\phi \left(\mathbf{x}_i - \mathbf{X}_j, h^\phi \right)}, \tag{4.21}$$

where $\phi_i = \phi(\mathbf{x}_i)$ is the CF at cell-center i with location $\mathbf{x}_i$. We have selected the same weight function W^a used to compute the pressure gradient ($W^\phi = W^a$), but have chosen a different influence radius of $h^\phi = 1.75\Delta x$, where Δx is the Eulerian grid spacing, to ensure the transition in CF is relatively independent of particle number density and smooth on the Eulerian grid. For a system with more than two phases, the color function is a vector quantity with a value for each phase ζ^α, computed according to

$$\phi_i^\alpha = \frac{\sum_{j=1}^{N_p} \chi_j^\alpha W_{ij}^\phi \left(\mathbf{x}_i - \mathbf{X}_j, h^\phi\right)}{\sum_{j=1}^{N_p} W_{ij}^\phi \left(\mathbf{x}_i - \mathbf{X}_j, h^\phi\right)}. \tag{4.22}$$

4.3.2 Implementation

A brief parametric study was performed to select the weights and coefficients in Eq. (4.20). The study quantified performance of the PMP method, as a function of the weights and coefficients, in a standard phase transport test problem. The parametric study is presented in Section 4.4.1.3 to illustrate model sensitivity. In this section, we simply outline which weights and coefficients were selected to close Eq. (4.20).

We select the quartic weight proposed by Lucy [87] for the pressure weight function W^a,

$$W^a(A) = \begin{cases} -3A^4 + 8A^3 - 6A^2 + 1, & 0 \leq A \leq 1, \\ 0, & A > 1, \end{cases} \tag{4.23}$$

where $A = |\mathbf{X}_i - \mathbf{X}_j| / h^a$. The normalization factor typically present in this quartic weight has been absorbed into the parameter c_1. The influence radius is selected as $h^a = 2.5 \left(V/N_{pc}\right)^{1/3}$, such that each particle has approximately 65 neighbors. The coefficient c_1 is determined empirically by considering an initially random distribution of particles and observing how they relax into a lattice using a fixed time step Δt. Using this procedure, we select $c_1 = 0.01 \left(V/N_{pc}\right)^{1/3} / \Delta t^2$. This couples the temporal behavior of the particle field to the Eulerian time step Δt. The weight function used

in the surface interactions between particles of the same phase, W^b, is given by

$$W^b(B) = \begin{cases} -\sin(\pi B)\cos(\pi B)\left(1 - B^{1.75}\right)^4, & 0 \leq B \leq 1, \\ 0, & B > 1, \end{cases} \tag{4.24}$$

where $B = |\mathbf{X}_i - \mathbf{X}_j| / h^b$ and $h^b = 1.9\left(V/N_{pc}\right)^{1/3}$. We have chosen the profile of W^b to be similar to the profiles proposed by Tartakovsky and Panchenko [133]: it generally produces an attractive force between particles of the same phase, but becomes a repulsive force if particles get too close to each other. This makes particles near the interface feel an attractive force toward their own phase, but prevents particle clustering [133]. The scaling coefficient for W^b is selected as $c_2 = 1.25c_1$. The weight function for surface interactions between difference phases, W^c, is selected as

$$W^c(A, B) = \begin{cases} -12A^3 + 24A^2 - 12A, & 0 \leq B \leq 1, \\ 0, & B > 1, \end{cases} \tag{4.25}$$

which is a repulsive term that depends on both h^a and h^b. The profile of W^c is always repulsive (it is the derivative of the pressure weight W^a), meaning particles near the interface feel a force that serves to move them away from particles of a different phase. The coefficient for this term is chosen to be $c_3 = 0.05c_1$. We use both W^b and W^c so that particles have two mechanisms to remain with particles of their own phase: they are attracted to particles of their own phase (provided they do not become too close to each other), and they are repulsed by particles of other phases. Because all particle interactions are constructed in an equal and opposite pair-wise fashion, the particle interaction terms discretely conserve momentum [133].

We also neglect the contribution from $\mathbf{u}_i'$ in Eq. (4.20), yielding the numerical model for the PMP velocity used in this work,

$$\mathbf{U}_i = \overline{\mathbf{u}}_i - \sum_{j=1}^{N_p} \left[c_1 \nabla W_{ij}^a + \left(c_2 W_{ij}^b + c_3 W_{ij}^c \right) \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \right] \Delta t. \tag{4.26}$$

The rational for neglecting $\mathbf{u}_i'$ is two-fold: inclusion of $\mathbf{u}_i'$ introduces stability considerations, whereas the residual acceleration terms have been designed to produce a stable particle field with constant number density for an arbitrary time step; additionally, neglecting $\mathbf{u}_i'$ reduces the amount of data communicated in a distributed memory computing system by nearly a factor of two and significantly decreases compute time. For compactness in upcoming notation, we introduce $\mathbf{a}_i$ as the residual particle acceleration, defined by

$$\mathbf{a}_i = -\sum_{j=1}^{N_p} \left[c_1 \nabla W_{ij}^a + \left(c_2 W_{ij}^b + c_3 W_{ij}^c \right) \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \right] \tag{4.27}$$

such that

$$\mathbf{U}_i = \overline{\mathbf{u}}_i + \mathbf{a}_i \Delta t. \tag{4.28}$$

The particle transport and Eulerian velocity are staggered in time and integrated with a two-step, explicit, iterative Crank-Nicholson scheme. An outline of the time integration procedure is shown in Fig. 4.1. The primary time advancement loop falls within the grey rectangle, while sub-processes are connected by dashed lines and appear at the bottom of the flow chart. Processes that involve the PMPs are red, processes that involve the Eulerian grid are green, and processes that involve both are yellow. The current time level of the PMP positions and the Eulerian velocity are shown between each significant process in the time advancement loop.

Time advancement of the particle field $\mathbf{X}_i(t)$ and the velocity field $\mathbf{u}(\mathbf{x}, t)$ proceeds as follows:

1. The initial condition is imposed, comprised of the particle positions at time level $n - 1/2$, $\mathbf{X}_i^{n-\frac{1}{2}}$, and the Eulerian velocity field at time level n, $\mathbf{u}^n$.

2. The simulation time is incremented by Δt and the iteration number n is incremented by 1.

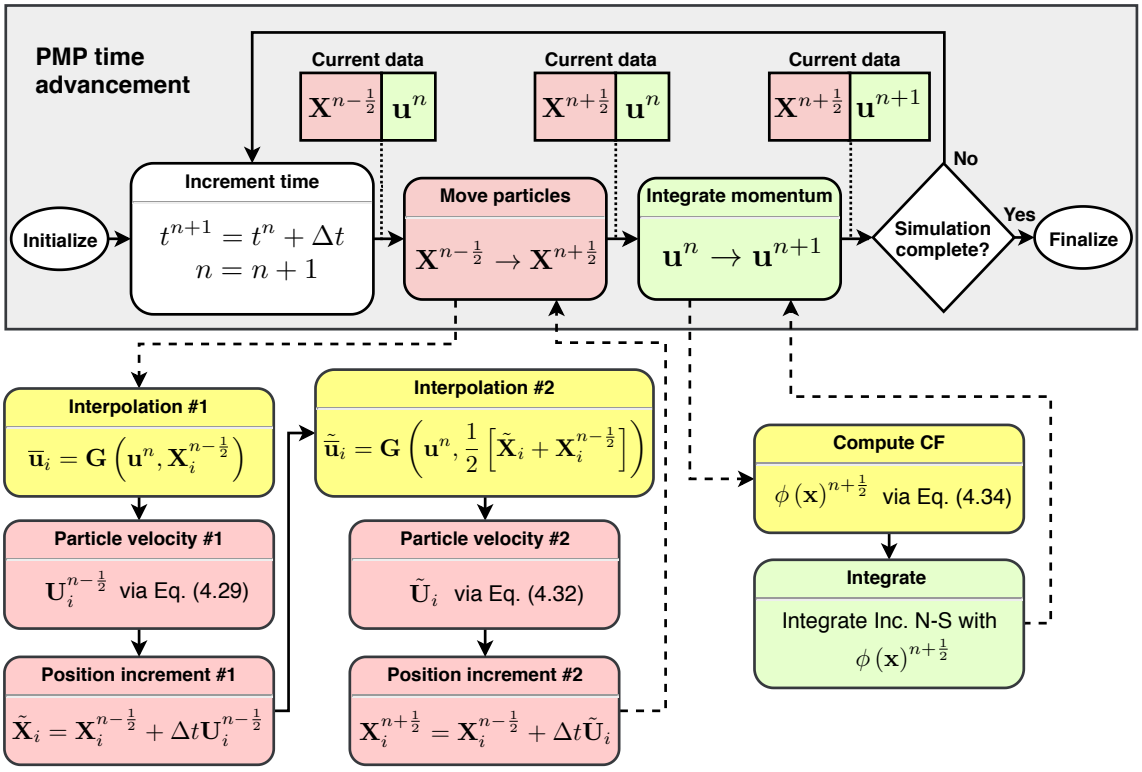

Figure 4.1. Flow chart of the time advancement procedure for the PMP method. PMP operations/data are shown in red, grid operations/data are shown in green, and mixed operations/data are shown in yellow. The color function in this flow chart assumes a two-phase system. (For interpretation of the colors in the figure, the reader is referred to the web version of this dissertation.)

3. The particle positions are advanced from $\mathbf{X}_i^{n-\frac{1}{2}}$ to $\mathbf{X}_i^{n+\frac{1}{2}}$ with a two-step, explicit, iterative Crank-Nicholson scheme described here in step #3. The current Eulerian velocity $\mathbf{u}^n$ is interpolated to the current particle positions $\mathbf{X}_i^{n-\frac{1}{2}}$, returning the interpolated particle velocity $\overline{\mathbf{u}}_i = \mathbf{G}\left(\mathbf{u}^n, \mathbf{X}_i^{n-\frac{1}{2}}\right)$. The PMP velocities are then computed according to Eq. (4.28),

$$\mathbf{U}_i^{n-\frac{1}{2}} = \overline{\mathbf{u}}_i + \mathbf{a}_i\left(\mathbf{X}_i^{n-\frac{1}{2}}, \mathbf{X}_j^{n-\frac{1}{2}}\right)\Delta t \tag{4.29}$$

and the particles are moved to an intermediate position $\tilde{\mathbf{X}}_i$,

$$\tilde{\mathbf{X}}_i = \mathbf{X}_i^{n-\frac{1}{2}} + \Delta t \mathbf{U}_i^{n-\frac{1}{2}}. \tag{4.30}$$

An intermediate interpolated velocity is then computed with the particles positioned at the midpoint of $\tilde{\mathbf{X}}_i$ and $\mathbf{X}_i^{n-\frac{1}{2}}$ according to

$$\tilde{\bar{\mathbf{u}}}_i = \mathbf{G}\left(\mathbf{u}^n, \frac{1}{2}\left[\tilde{\mathbf{X}}_i + \mathbf{X}_i^{n-\frac{1}{2}}\right]\right). \tag{4.31}$$

The PMP velocity is then computed using $\tilde{\bar{\mathbf{u}}}_i$ and the particles positioned at the midpoint of $\tilde{\mathbf{X}}_i$ and $\mathbf{X}_i^{n-\frac{1}{2}}$ according to

$$\tilde{\mathbf{U}}_i = \tilde{\bar{\mathbf{u}}}_i + \mathbf{a}_i\left(\frac{1}{2}\left[\tilde{\mathbf{X}}_i + \mathbf{X}_i^{n-\frac{1}{2}}\right], \frac{1}{2}\left[\tilde{\mathbf{X}}_j + \mathbf{X}_j^{n-\frac{1}{2}}\right]\right)\Delta t \tag{4.32}$$

The particles then advance to their position at time level $n+\frac{1}{2}$ according to

$$\mathbf{X}_i^{n+\frac{1}{2}} = \mathbf{X}_i^{n-\frac{1}{2}} + \Delta t\tilde{\mathbf{U}}_i. \tag{4.33}$$

4. The Eulerian velocities are advanced from $\mathbf{u}^n$ to $\mathbf{u}^{n+1}$ using the phase information provided by the particles at the intermediate time $n+\frac{1}{2}$. The Eulerian CF is computed at each cell center i according to

$$\phi_i^{n+\frac{1}{2}} = \frac{\sum_{j=1}^{N_p}\chi_j W_{ij}^{\phi}\left(\mathbf{x}_i - \mathbf{X}_j^{n+\frac{1}{2}}, h^{\phi}\right)}{\sum_{j=1}^{N_p} W_{ij}^{\phi}\left(\mathbf{x}_i - \mathbf{X}_j^{n+\frac{1}{2}}, h^{\phi}\right)}. \tag{4.34}$$

(Note that the indicator of particle j, χ_j, is not associated with a time level because it is time invariant.) If there are more than two phases in the system, the CF is instead computed for every phase according to Eq. (4.22). The fluid properties are then computed as a function of $\phi_i^{n+\frac{1}{2}}$ (or $\phi_i^{\alpha n+\frac{1}{2}}$ for a system with more than two phases) and the incompressible Navier-Stokes equations are integrated from time level n to $n+1$.

5. If the desired simulation time has been reached, the simulation is complete. If not, the process begins again at step #2.

4.4 Performance assessment

We assess performance by solving a number of phase transport problems and by simulating a variety of multiphase fluid flows. The former uses prescribed Eulerian velocity fields while the latter acquires the Eulerian velocity through numerical solution of the Navier-Stokes equations. In performing these simulations we establish performance characteristics associated with resolution (Eulerian mesh resolution and Lagrangian particle number density), and demonstrate the suitability of the approach for simulating real flow problems.

4.4.1 Phase transport

4.4.1.1 Single vortex problem

We consider the single vortex problem [76] as a first assessment of the phase transport performance of the approach. This problem has been used extensively to test VOF methods [104, 106] and particle-based methods, including the Lagrangian level set [56]. The problem involves placing a thin cylinder of phase ζ^1 with radius $r_o = 0.15$ centered at position $x = 0.5$, $y = 0.75$ in a domain of extent $x = \{0, 1\}$, $y = \{0, 1\}$, and $z = \{-1.5\Delta x, 1.5\Delta x\}$. The domain is otherwise filled with phase ζ^2. The cylinder is then deformed by a prescribed velocity field

$$\mathbf{u}(x, y, z) = 2 \begin{bmatrix} -\sin^2(\pi x)\sin(\pi y)\cos(\pi y)\cos(\pi t/T) \\ \sin^2(\pi y)\sin(\pi x)\cos(\pi x)\cos(\pi t/T) \\ 0 \end{bmatrix}, \tag{4.35}$$

where t is the time and $T = 8$ is the period of the vortex [56].

Numerical specifications

The Eulerian mesh is defined by $N_x = N_y$ grid points in the x and y directions and $N_z = 3$ grid points in the z direction. The thickness in z is always three mesh spacings. A grid comprised of a $64 \times 64 \times 3$ matrix of nodes is referred to as an $N_x = 64$ grid. All three spatial directions are treated periodically and a constant time-step is evaluated using the initial velocity field,

$$\Delta t = 0.8 \times \min \left(\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} \right)^{-1} \Bigg|_{t=0}. \tag{4.36}$$

The time-step is a function of the Eulerian mesh and velocity field and is independent of the particle field. It is evaluated once, at time $t = 0$, and the global minimum is used for the duration of the simulation.

In each Eulerian control volume, N_{pc} particles are seeded at random locations. Each particle initially belongs to phase ζ^2 with $\chi_i = 0$. After the particles are seeded, the initial particle distribution is relaxed for 100 iterations using the time-step Δt defined by Eq. (4.36). The relaxation is performed by setting the Eulerian velocity field temporarily to zero, and then integrating the phase transport equations for 100 time-steps. This results in a well-distributed particle field with approximately uniform number density. All PMPs located within a radius $r_o = 0.15$ of position $x = 0.5, y = 0.75$ are assigned a phase-indicating value $\chi_i = 1$ and all other particles are assigned $\chi_i = 0$. An additional 100 iterations of particle relaxation are performed, and the resultant particle field is considered to be the initial condition at time $t = t_o$. This particle relaxation procedure is used for phase initialization throughout this work.

Shape evolution

Figure 4.2 shows results of the single vortex problem for a mesh resolution of $N_x = 256$ and a particle number density of $N_{pc} = 4$ particles per Eulerian cell. Red PMPs

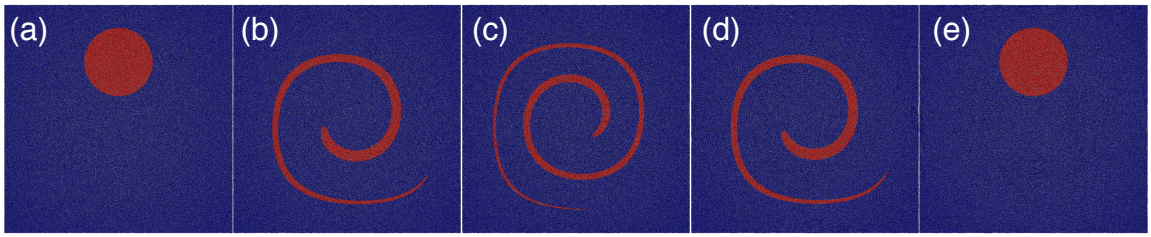

Figure 4.2. Evolution of the particle field in the single vortex problem with $N_{pc} = 4$ and $N_x = 256$ from time $t = 0$ (left) to time $t = 8$ (right).

corresponding to phase ζ^1 and blue PMPs corresponding to phase ζ^2 appear between time $t = 0$ and $t = 8$. The velocity field deforms the red circle into a long ligament by $t = 1$. Deformation continues until a thin spiral is achieved at $t = 4$. The flow reverses direction at $t = 4$ and the initial circle of phase ζ^1 is reconstituted by time $t = 8$. The PMPs remain uniformly distributed for all times. The phases also remain contiguous and distinctly separated, meaning single particles of phase ζ^1 do not become surrounded by phase ζ^2 (and vice versa). The large scale motion of the PMPs is determined by the Eulerian velocity. Maintenance of a well-distributed and immiscible particle field is accomplished by the particle acceleration terms present in Eq. (4.28).

Accuracy is assessed via Eulerian shape error and Eulerian volume conservation error. Shape error quantifies spatial deviation of the final CF distribution ($t = 8$) from the initial CF distribution ($t = 0$) [86, 47, 106]. The shape error ΔS is defined by

$$\Delta S = \frac{V}{3\Delta x} \sum_{n=1}^{N_{cv}} \left| \phi_n(t) - \phi_n(t_o) \right| \Big|_{t=8}, \tag{4.37}$$

where $N_{cv} = 3N_x^2$ is the number of control volumes with volume $V = \Delta x^3$ in the mesh, $\phi_n(t)$ is the CF of the n^{th} control volume at time t, $\phi_n(t_o)$ is the initial CF of the n^{th} control volume, and $1/3\Delta x$ is a normalization factor used to ensure the three-node thickness in z does not artificially influence the convergence properties.

Conservation error considers the change in total Eulerian CF [86, 47, 106]. It is a

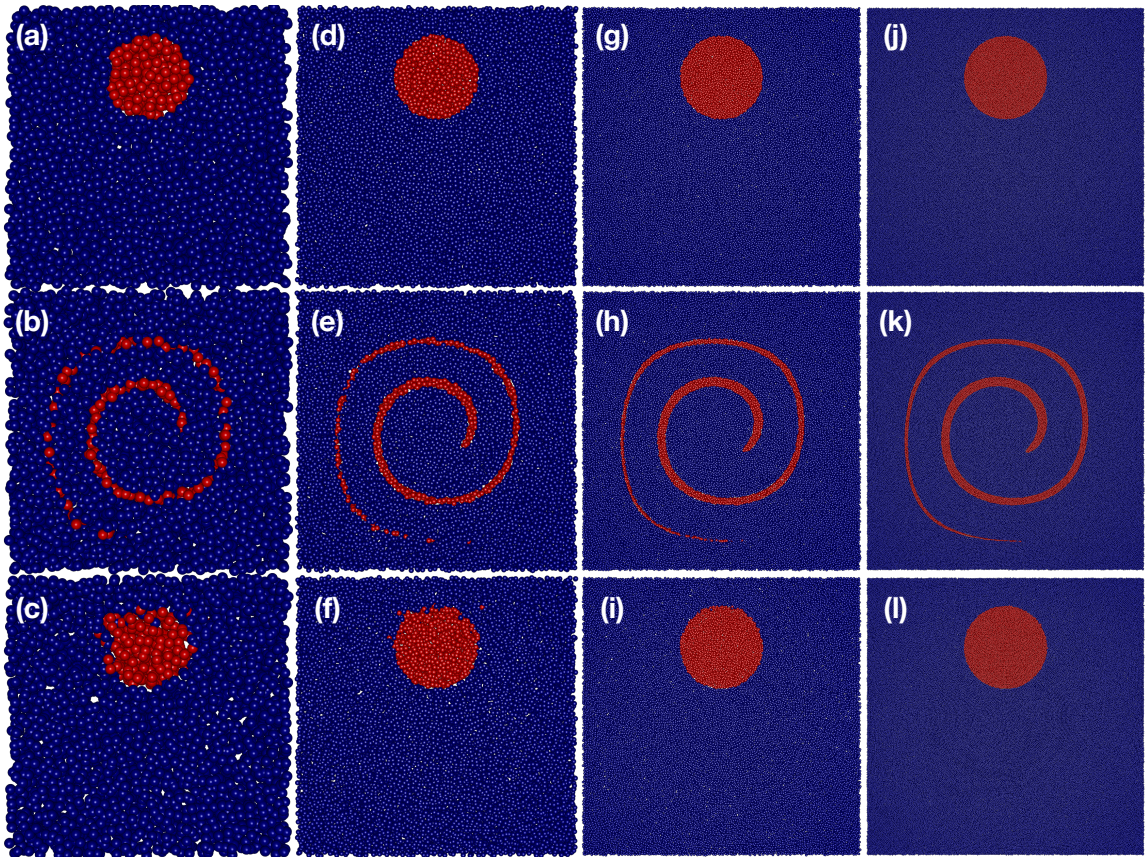

Figure 4.3. Particle distributions in the single vortex problem at times $t = 0$ (top row), $t = 4$ (middle row), and $t = 8$ (bottom row) as a function of mesh resolution N_x with $N_{pc} = 4$ particles per cell: panels *a-c*, $N_x = 16$; panels *d-f*, $N_x = 32$; panels *g-i*, $N_x = 64$; and panels *j-i*, $N_x = 128$.

signed percent error defined by

$$\Delta\phi^{\star}(t) = 100\left(\frac{\sum_{n=1}^{N_{cv}}\phi_n(t)}{\sum_{n=1}^{N_{cv}}\phi_n(t_o)} - 1\right). \tag{4.38}$$

It should be noted that, while the total Eulerian CF may vary, the total mass of each phase remains constant in the Lagrangian space.

Effects of grid resolution

We perform simulations with $N_{pc} = 4$ particles per Eulerian cell at four grid resolutions: $N_x = 16$, $N_x = 32$, $N_x = 64$, and $N_x = 128$. The resulting particle fields are

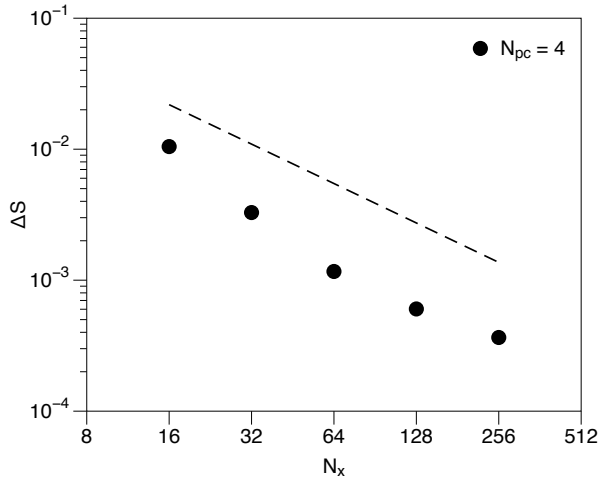

Figure 4.4. Convergence of the shape error ΔS in the single vortex problem as a function of mesh resolution N_x with $N_{pc} = 4$ particles per cell. The dashed line shows first-order convergence.

shown in Fig. 4.3. Higher grid resolutions better approximate the circle at the initial time; the lowest resolution grid with $N_x = 16$ shows obvious defects, while the highest resolution grid with $N_x = 128$ is well–defined. At time $t = 4$, the elongated ligament breaks into small pieces at low resolutions ($N_x = 16$ and $N_x = 32$), but remains well-defined at the higher resolutions ($N_x = 64$ and $N_x = 128$). At time $t = 8$, the lowest resolutions of $N_x = 16$ and $N_x = 32$ present a moderate amount of dispersion between the red and blue phases. This unphysical behavior is significantly reduced at a resolution of $N_x = 64$, and absent at $N_x = 128$. Convergence of the shape error is shown in Fig. 4.4. (The result for the $N_x = 256$ simulation is also included.) Over the range of resolutions considered, a first-order convergence rate is observed for the shape error ΔS with respect to increasing the mesh resolution N_x.

The CF conservation error is shown in Fig. 4.5. The temporal evolution is shown in Fig. 4.5 panel *a* and convergence behavior of the maximum error is shown in Fig. 4.5 panel *b*. The fluctuations in $\Delta\phi^\star$ occur with high frequency. At the lowest resolution ($N_x = 16$), a transient error of greater than 0.7% appears at time $t \approx 7$. As the mesh resolution increases from $N_x = 16$ to $N_x = 256$, the maximum fluctuation decreases

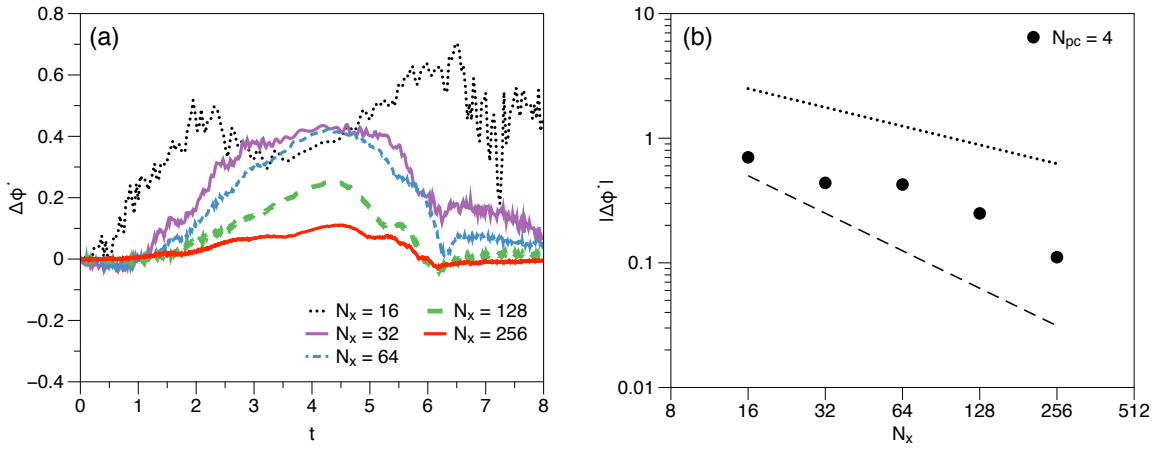

Figure 4.5. Percent error in CF conservation in the single vortex problem as a function of mesh resolution N_x with $N_{pc} = 4$ particles per cell: (a) transient evolution of percent error $\Delta\phi^\star$; (b) magnitude of maximum percent error $|\Delta\phi^\star|$, where the dashed line shows first-order convergence and the dotted line shows convergence with the square root.

to approximately 0.1%. Figure 4.5 panel *b* shows that the maximum conservation error decreases approximately with the root of the mesh resolution over the range considered.

Effects of particle number density

We also assess phase tracking as a function of particle number density N_{pc}. Using Eulerian grids with resolutions of $N_x = 16$ and $N_x = 32$, we perform simulations with $N_{pc} = 2$, $N_{pc} = 8$, $N_{pc} = 16$, and $N_{pc} = 64$ particles per cell. The results for $N_x = 32$ are shown in Fig. 4.6. Increasing the particle number density results in a better approximation of the initial cylinder at time $t = 0$, similar to the behavior observed when increasing the Eulerian mesh resolution. Increasing the particle number density also improves the connectivity of the elongated ligament at the intermediate time $t = 4$. While the $N_{pc} = 2$ case does not yield a contiguous thin filament, the $N_{pc} = 64$ case preserves connectivity in all locations excluding the trailing edge. Higher particle

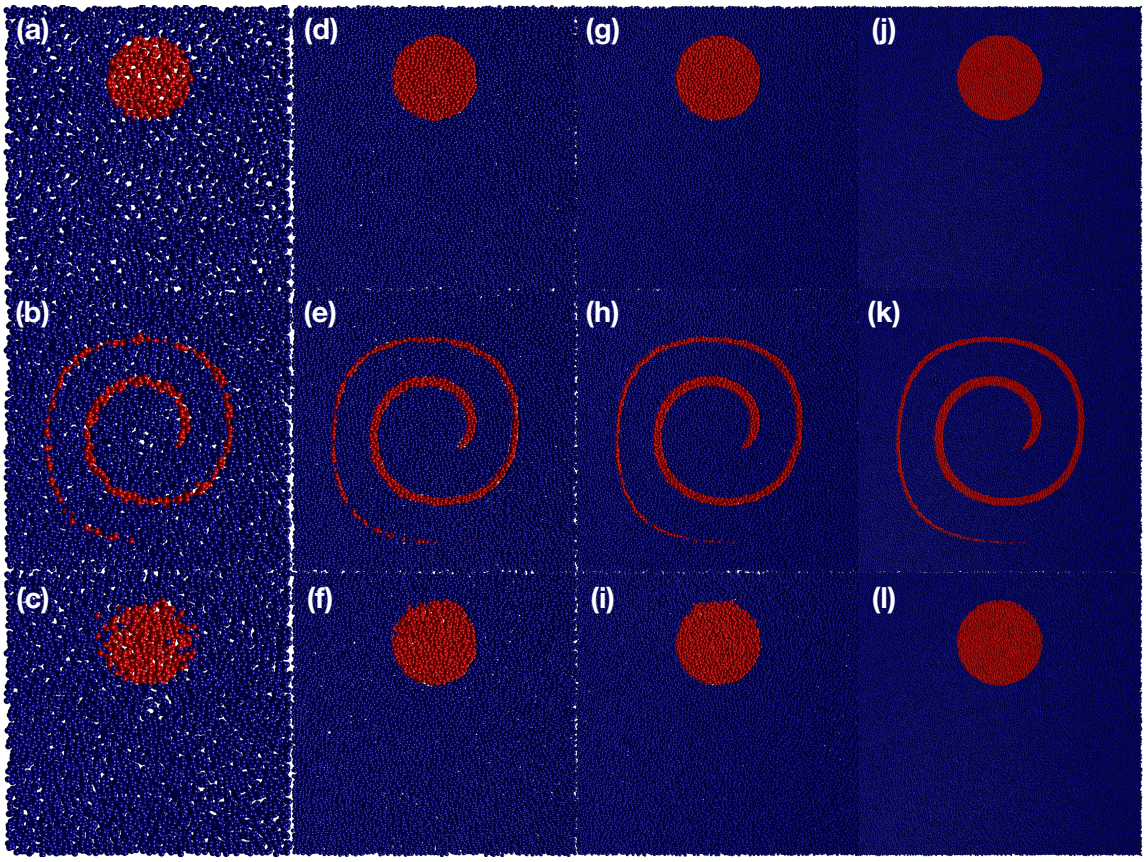

Figure 4.6. Particle distributions in the single vortex problem at times $t = 0$ (top row), $t = 4$ (middle row), and $t = 8$ (bottom row) as a function of the number of particles per cell N_{pc} for a mesh resolution of $N_x = 32$: panels *a-c*, $N_{pc} = 2$; panels *d-f*, $N_{pc} = 8$; panels *g-i*, $N_{pc} = 16$; and panels *j-i*, $N_{pc} = 64$.

number densities yield reduced inter-phase dispersion and a better representation of the expected circular shape at time $t = 8$.

Shape error convergence is shown as a function of N_{pc} in Fig. 4.7 for grid resolutions of $N_x = 32$ and $N_x = 16$. The shape error converges with approximately first order for both grid resolutions over the range of particle number densities considered. The results demonstrate that increasing either the particle number density N_{pc} or the grid resolution N_x decreases the shape error.

Transient fluctuations in the total Eulerian CF as a function of particle number

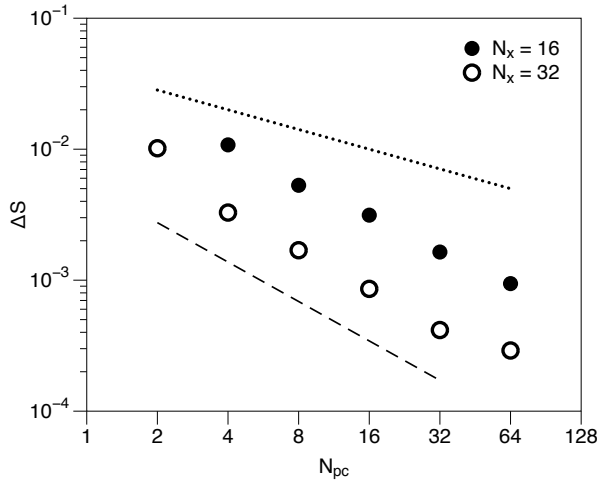

Figure 4.7. Convergence of the shape error ΔS in the single vortex problem as a function of the number of particles per cell N_{pc} at mesh resolutions of $N_x = 16$ and $N_x = 32$. The dashed line shows first-order convergence and the dotted line shows convergence with the square root.

density are presented in Fig. 4.8. The results for the $N_x = 16$ simulations are shown in Fig. 4.8 panel *a* and those for the $N_x = 32$ simulations are shown in Fig. 4.8 panel *b*. In all cases, the maximum magnitude of CF conservation error is less than 0.7%. However, as the particle number density increases from $N_{pc} = 4$ to $N_{pc} = 64$, the maximum transient error does not markedly decrease. In fact, the results suggest that incrementally increasing the particle number density may increase the maximum CF error by a small amount.

4.4.1.2 Five phase transport

The methodology explicitly tracks parcels of fluid and can therefore solve for the evolution of any number of phases without modification. We use the single vortex problem to consider the transport of five phases. We introduce four immiscible circles located symmetrically on the top, left, right, and bottom of the domain, as shown in Fig. 4.9 panel *a*. The grid resolution is $N_x = 64$ and the particle number density is $N_{pc} = 8$. We require a method to visualize each phase, so we simply visualize a

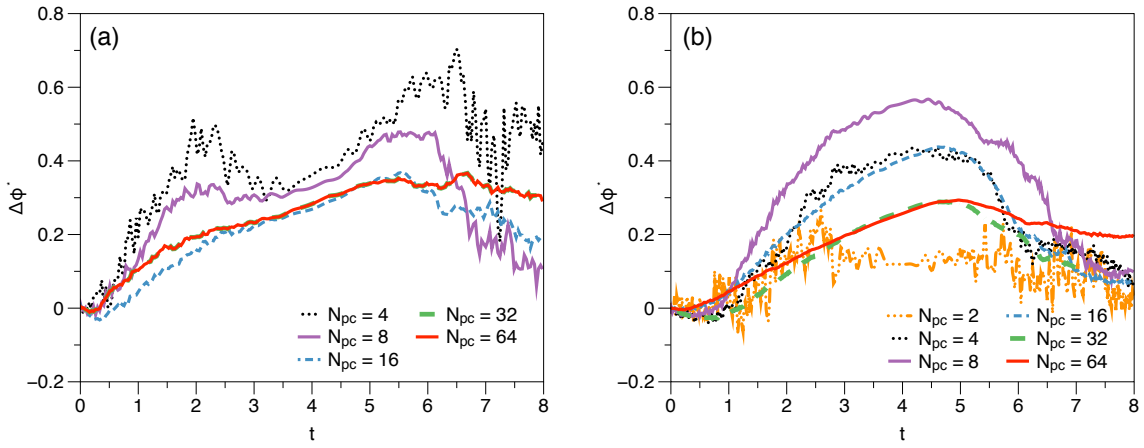

Figure 4.8. Transient percent error in CF conservation in the single vortex problem as a function of the number of particles per cell N_{pc} at mesh resolutions of (a) $N_x = 16$ and (b) $N_x = 32$.

density in Fig. 4.9 in by assigning $\rho^1 = 1$, $\rho^2 = 2$, $\rho^3 = 3$, $\rho^4 = 4$, and $\rho^5 = 5$. The dark blue phase corresponds to ζ^1, the light blue phase corresponds to ζ^2, the green phase (background fluid) corresponds to ζ^3, the orange phase corresponds to ζ^4, and the dark red phase corresponds to ζ^5.

Figure 4.9 shows the single vortex problem performed with five phases. Panels *a*–*e* show the time evolution of the PMPs, while panels *f*–*j* show the time evolution of the Eulerian field. The PMPs corresponding to the four dispersed phases transition from the initial condition in panel *a* to a set of four symmetric ligaments by time 0.7 in panel *b*. At the intermediate time $t = 4$ in panel *c*, all five phases have formed well-defined, thin ligaments in the center of the domain. At times $t = 7.7$ and $t = 8$ in panels *d* and *e*, each phase returns to its initial position. Panels *f*–*j* show the corresponding Eulerian field at the same times. The boundary of each phase is clearly defined in the initial condition shown in panel *f*. By time $t = 0.7$ in panel *g*, the phase boundaries approach each other on the $N_x = 64$ grid. At the intermediate time of $t = 4$ in panel *h*, the boundaries of each phase have become blurred on the Eulerian grid, but remain well-defined by the PMPs in panel *c*. Because the PMPs retain the small scale phase

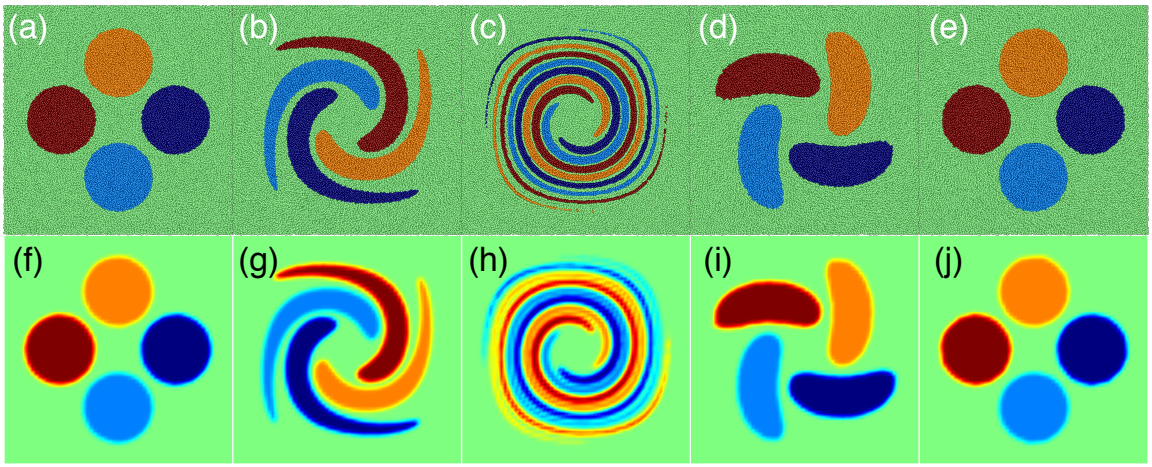

Figure 4.9. Particle distributions (top row) and Eulerian field (bottom row) in the single vortex problem with five phases. The times shown, increasing from left to right, are $t = 0$, $t = 0.7$, $t = 4$, $t = 7.7$, and $t = 8$. The Eulerian grid resolution is $N_x = 64$ and the particle number density is $N_{pc} = 8$ particles per cell. (For interpretation of the colors in the figure, the reader is referred to the web version of this dissertation.)

information, the Eulerian field reassumes well–defined boundaries in panels i and j. The PMPs have the ability to preserve phase information at scales smaller than those resolvable by the Eulerian grid.

4.4.1.3 Effects of model parameters

The PMP scheme involves a large number of model parameters. Changing the parameters defined in Section 4.3 changes the performance of the method. Here we examine the sensitivity of the method to variations in some of the model parameters. The surface tension acceleration $\mathbf{F}'_i$ depends on four model parameters: the weight functions W^b and W^c, and their respective coefficients c_2 and c_3. Table 4.1 defines the parameters of five different PMP implementations, denoted by PMP #1-#5. PMP #1 is the implementation presented and evaluated thus far and defined in Section 4.3.2. All parameters not specified in Table 4.1 for PMP #2-#5 are the same as for PMP #1 defined in Section 4.3.2.

Table 4.1. Model parameters for the residual acceleration $\mathbf{F}'_i$ for five different PMP implementations, denoted by PMP #1 - #5.

PMP #	W_b	h_b	c_2	c_3
1	$\begin{cases} -\sin(\pi B)\cos(\pi B)\left(1-B^{1.75}\right)^4, & 0 \le B \le 1, \\ 0, & B > 1, \end{cases}$	$1.9\,(V/N_{pc})^{1/3}$	$1.25c_1$	$0.05c_1$
2	$\begin{cases} -\sin(\pi B)\cos(\pi B)\left(1-B^{6}\right)^4, & 0 \le B \le 1, \\ 0, & B > 1, \end{cases}$	$1.75\,(V/N_{pc})^{1/3}$	$1.25c_1$	$0.05c_1$
3	$\begin{cases} -\sin(\pi B)\cos(\pi B)\left(1-B^{1.75}\right)^3, & 0 \le B \le 1, \\ 0, & B > 1, \end{cases}$	$2.5\,(V/N_{pc})^{1/3}$	$0.2c_1$	$0.05c_1$
4	$\begin{cases} -\sin(\pi B)\cos(\pi B)\left(1-B^{1.75}\right)^4, & 0 \le B \le 1, \\ 0, & B > 1, \end{cases}$	$1.9\,(V/N_{pc})^{1/3}$	$1.25c_1$	0.0
5	$\begin{cases} -\sin(\pi B)\cos(\pi B)\left(1-B^{6}\right)^4, & 0 \le B \le 1, \\ 0, & B > 1, \end{cases}$	$1.75\,(V/N_{pc})^{1/3}$	$1.25c_1$	0.0

PMP #2 has an 8% narrower influence radius h_b than PMP #1, as well as a different weight profile W_b. PMP #3 has a 43% wider influence radius h_b, a different weight profile W^b, and an 84% smaller weight coefficient c_2. Figure 4.10 shows the performance of PMP #1-#3 in the single vortex problem. Panel *a* shows convergence of the shape error ΔS and panel *b* shows the magnitude of the maximum percent error in CF conservation $|\Delta\phi^\star|$. Differences in shape error for all three implementations are less than a factor of three for the range of resolutions considered. PMP #2 produces the largest errors at high resolutions. PMP #1 and #3 produce similar errors over the range of resolutions tested. With respect to conservation error, PMP #1 out-performs the other implementations over the entire range of resolutions.

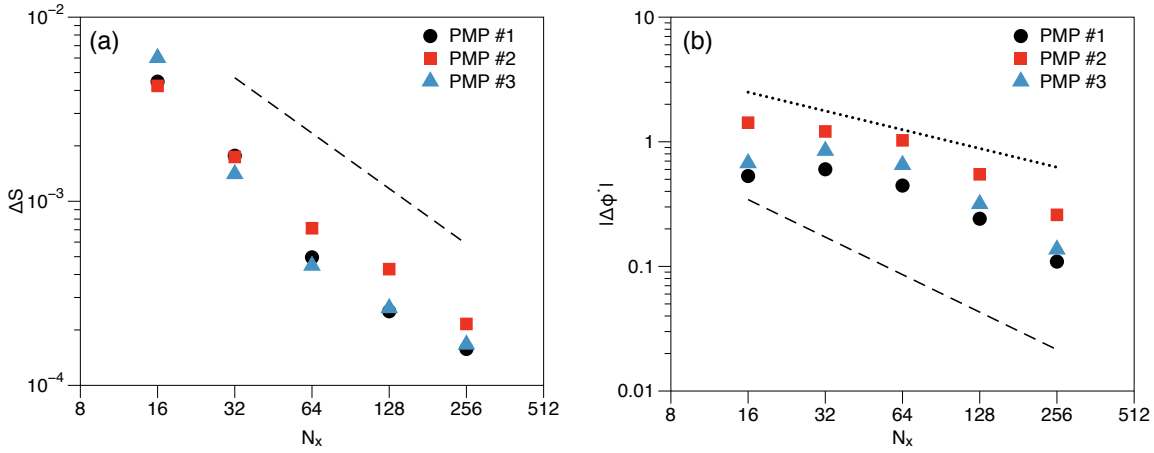

Figure 4.10. Comparison of PMP implementations #1, #2, and #3 in the single vortex problem: (a) convergence of Shape error and (b) magnitude of maximum percent error in CF conservation $|\Delta\phi^{\star}|$. The dashed line shows first-order convergence and the dotted line shows convergence with the square root.

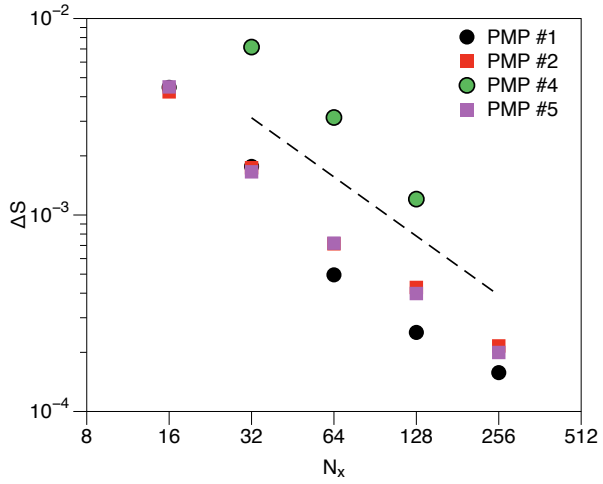

Figure 4.11. Comparison of PMP implementations #1, #2, #4, and #5 with respect to shape error in the single vortex problem. The dashed line shows first-order convergence.

PMP #4 is the same as PMP #1, except the coefficient responsible for surface interactions between particles of different phases, c_3, is set to $c_3 = 0$. Similarly, PMP #5 is the same as PMP #2, but with $c_3 = 0$. Figure 4.11 shows the shape error performance of PMP #1, #2, #4, and #5 in the single vortex problem. In the case of PMP #2 and #5, the repulsive component of surface tension does not play a large factor. PMP #2 and #5 perform similarly for all cases tested. PMP #4 produces an order of magnitude larger error than the other methods. This is because the attractive component of surface tension (W^b and its coefficient c_2) is insufficient for keeping the phases distinctly separated – for PMP #1, the repulsive component of surface tension, produced by W^c and c_3, is required for good performance. The remainder of the analyses and assessments performed in this work utilizes PMP #1, which is referred to simply as the PMP method for brevity.

4.4.1.4 Three-dimensional vortex problem

The three-dimensional vortex problem is used as a second evaluation platform [76]. This problem, like the single vortex problem, has been widely used to evaluate the performance of phase transport schemes [104, 106]. The problem involves placing a sphere of phase ζ^1 with radius $r_o = 0.15$ at position $x = 0.35$, $y = 0.35$, $z = 0.35$ in a domain of extent $x = \{0, 1\}$, $y = \{0, 1\}$, and $z = \{0, 1\}$. The domain is otherwise filled with phase ζ^2. The sphere is then deformed by a prescribed velocity field

$$\mathbf{u}(x, y, z) = \begin{bmatrix} 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\cos(\pi t/T) \\ -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)\cos(\pi t/T) \\ -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)\cos(\pi t/T) \end{bmatrix}, \tag{4.39}$$

where t is the time and $T = 3$ is the period of the vortex.

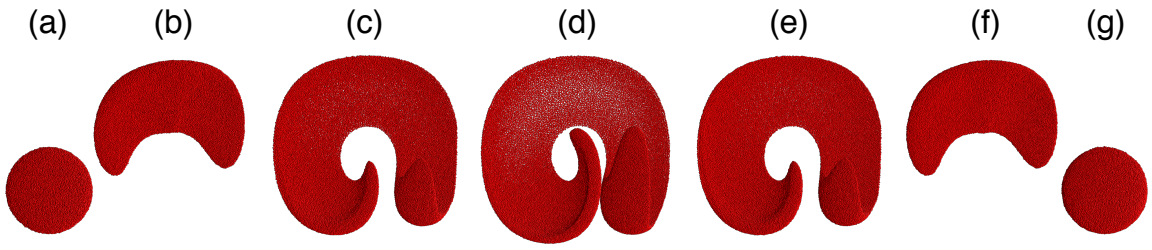

Figure 4.12. Temporal evolution of the particle field from $t = 0$ (panel a) to $t = 3$ (panel g) in the 3D deformation test for a grid resolution of $N_x = 128$ with $N_{pc} = 4$ particles per cell.

Numerical specifications

The time step is determined in the same manner as the single vortex problem. The computational grid is defined as having N_x nodes in all three spatial dimensions. Simulations are performed at 5 different grid resolutions: $N_x = 16$, $N_x = 32$, $N_x = 64$, $N_x = 128$, and $N_x = 256$. Simulations are also performed at six different particle number densities: $N_{pc} = 2$, $N_{pc} = 4$, $N_{pc} = 8$, $N_{pc} = 16$, $N_{pc} = 32$, and $N_{pc} = 64$.

Having adopted a fully three-dimensional domain, we require a new definition for shape error,

$$\Delta S = V \sum_{n=1}^{N_{cv}} \left| \phi_n(t) - \phi_n(t_o) \right| \Big|_{t=3}, \tag{4.40}$$

and, for completeness, the conservation error is re-introduced,

$$\Delta\phi^{\star}(t) = 100 \left(\frac{\sum_{n=1}^{N_{cv}} \phi_n(t)}{\sum_{n=1}^{N_{cv}} \phi_n(t_o)} - 1 \right). \tag{4.41}$$

Shape evolution

Figure 4.12 shows results of the three-dimensional vortex problem for $N_x = 128$ and $N_{pc} = 4$. The Red PMPs corresponding to phase ζ^1 are shown between times $t = 0$ and $t = 3$. The velocity field displaces and thins the sphere into a sheet by $t = 0.5$. The sheet continues to thin and begins to curl in all three dimensions until the intermediate time of $t = 1.5$. The velocity field reverses direction, and the particles

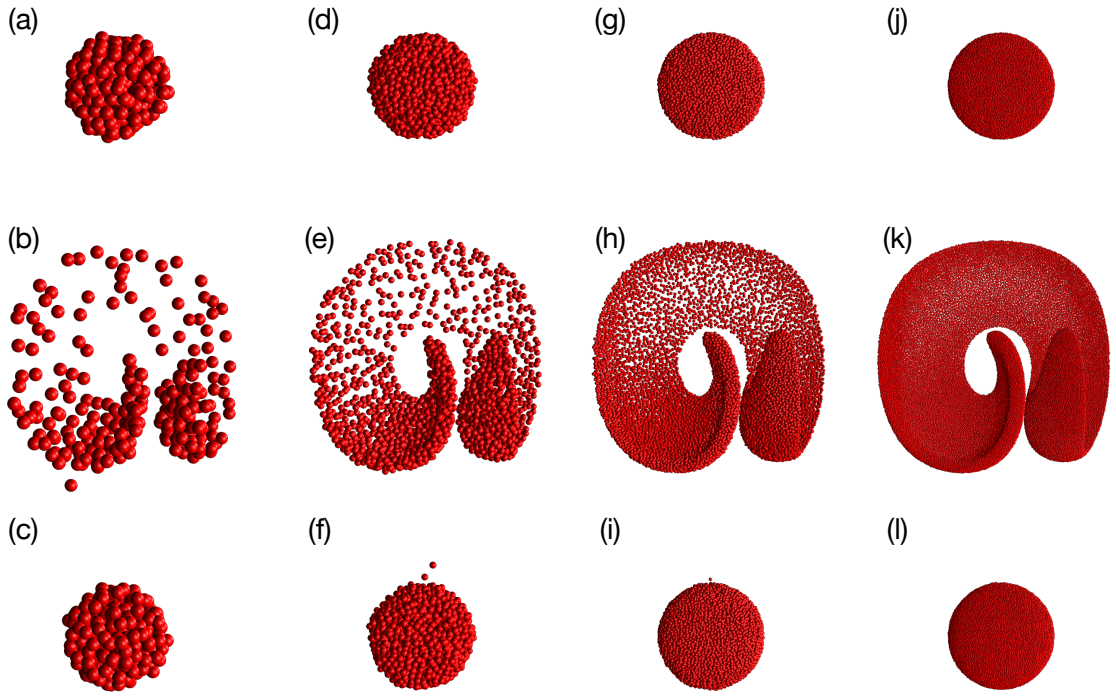

Figure 4.13. Particle distributions in the 3D deformation test at times $t = 0$ (top row), $t = 1.5$ (middle row), and $t = 3$ (bottom row) as a function of grid resolution N_x with $N_{pc} = 4$ particles per cell: panels *a-c*, $N_x = 16$; panels *d-f*, $N_x = 32$; panels *g-i*, $N_x = 64$; and panels *j-i*, $N_x = 128$.

reassume the initial spherical shape by $t = 3$. Much like the results for the single vortex problem, the particles remain uniformly distributed because of the particle acceleration terms.

Effects of grid resolution

The effects of grid resolution are demonstrated by performing simulations with a particle number density of $N_{pc} = 4$ particles per cell at four grid resolutions: $N_x = 16$, $N_x = 32$, $N_x = 64$, and $N_x = 128$. The resulting particle fields are shown in Fig. 4.13. Higher grid resolutions better describe the initial sphere; the lowest resolution grid with $N_x = 16$ has defects, while the highest resolution grid with $N_x = 128$ presents

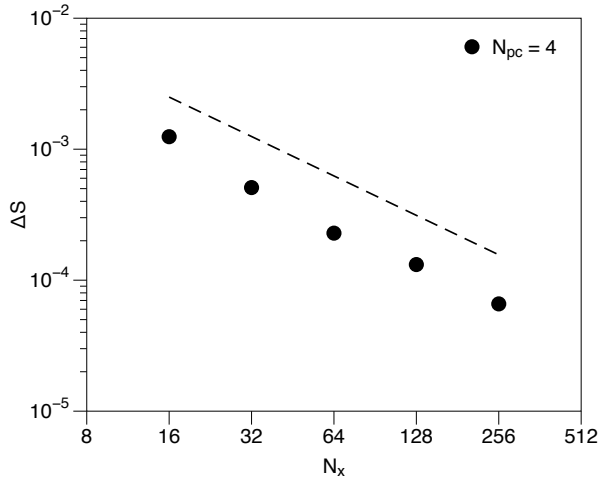

Figure 4.14. Convergence of the shape error ΔS in the 3D deformation test as a function of grid resolution N_x with $N_{pc} = 4$ particles per cell. The dashed line shows first-order convergence.

a well-defined sphere. At the intermediate time of $t = 1.5$, the thin film is not well-described by uniformly spaced red particles at low resolutions ($N_x = 16$ and to a lesser extent $N_x = 32$) but remains well-defined at the higher resolutions ($N_x = 64$ and $N_x = 128$). At the final time of $t = 3$, the lowest resolution cases ($N_x = 16$ and $N_x = 32$) deviate from a spherical shape and include dispersion between the phases (indicated by detached red particles). These errors are effectively eliminated in the $N_x = 64$ solution, which presents one detached particle, and superior performance is demonstrated by the $N_x = 128$ solution.

Convergence of shape error, ΔS, is shown in Fig. 4.14. (The result for an $N_x = 256$ simulation is also included.) Over the range of mesh resolutions considered, the results show an approximately first-order convergence rate for the shape error with respect to increasing mesh resolution N_x.

The CF conservation error is shown in Fig. 4.15. The temporal evolution is shown in Fig. 4.15 panel *a* and the convergence behavior of the maximum error is shown in Fig. 4.15 panel *b*. A maximum transient error of approximately 0.6% appears in the $N_x = 32$ simulation. Although the $N_x = 16$ simulation presents smaller errors than

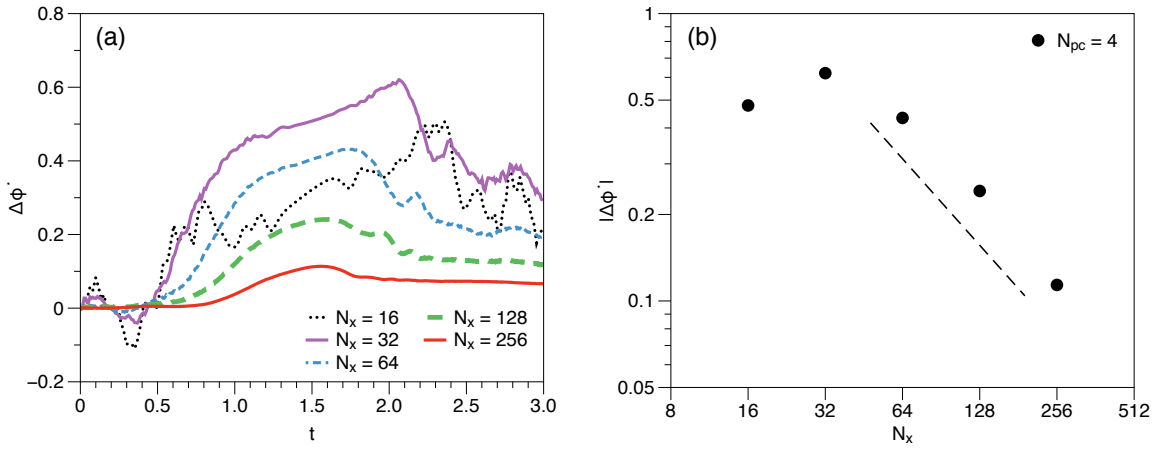

Figure 4.15. Percent error in CF conservation in the 3D deformation test as a function of grid resolution N_x with $N_{pc} = 4$ particles per cell: (a) transient evolution of percent error $\Delta\phi^\star$; (b) magnitude of maximum percent error $|\Delta\phi^\star|$. The dashed line in panel b shows first-order convergence.

the $N_x = 32$ simulation, errors generally decrease with increasing mesh resolution. As demonstrated in Fig. 4.15 panel b, the maximum conservation error decreases with approximately first-order over the range of resolutions tested.

Effects of particle number density

Phase tracking in the three-dimensional vortex problem is also assessed as a function of particle number density N_{pc}. We perform simulations with particle number densities of $N_{pc} = 2$, $N_{pc} = 8$, $N_{pc} = 16$, and $N_{pc} = 64$ at grid resolutions of $N_x = 16$, $N_x = 32$ and $N_x = 64$. The resulting particle fields at the final time $t = 3$ are shown in Fig. 4.16. Cases with low mesh resolution and low particle number density, such as panel a ($N_x = 16$ and $N_{pc} = 2$), produce non-spherical shapes at $t = 3$. Increasing the particle number density at low resolutions, as in panel d ($N_x = 16$ and $N_{pc} = 64$), results in a higher fidelity sphere, but does not eliminate dispersion between phases (detached red particles). Once a sufficiently high mesh resolution is

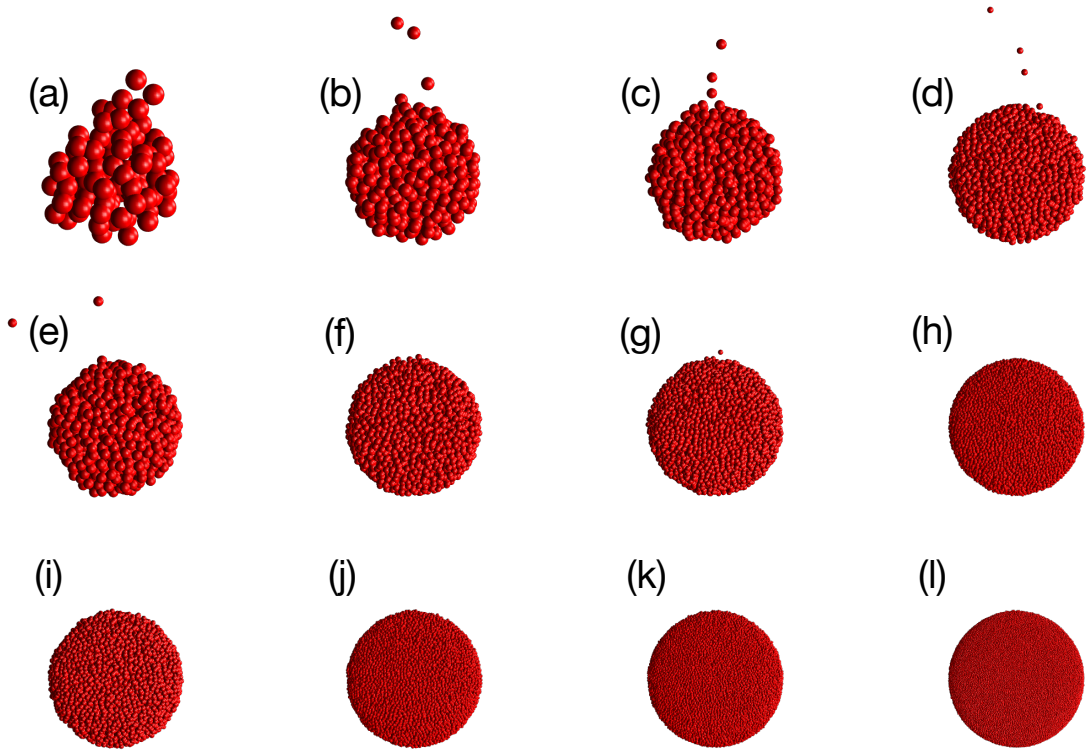

Figure 4.16. Particle fields in the 3D deformation test shown at the final time of $t = 3$ as a function of grid resolution N_x and particle number density N_{pc}. Particle number density is constant in each column, increasing from $N_{pc} = 2$, $N_{pc} = 8$, $N_{pc} = 16$, and $N_{pc} = 64$ from left to right. Grid resolution is constant in each row, increasing from $N_x = 16$, $N_x = 32$, and $N_x = 64$ from top to bottom.

achieved, high particle number densities are not necessarily required for good performance, as demonstrated by panel i ($N_x = 64$ and $N_{pc} = 2$). Increasing the particle number density on a high resolution mesh improves the fidelity of the final sphere, as in panel l ($N_x = 64$ and $N_{pc} = 64$).

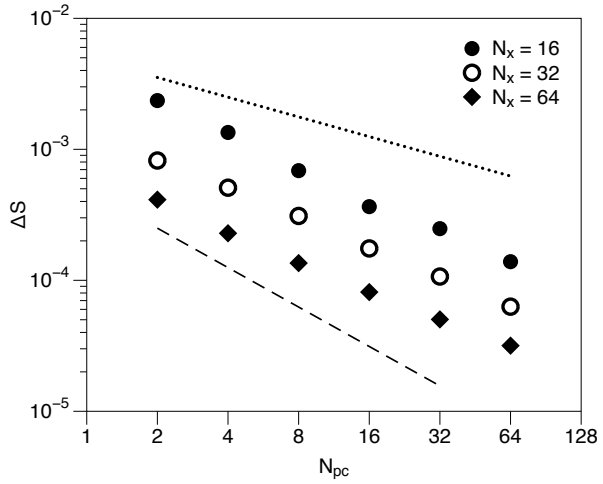

Figure 4.17. Convergence of the shape error ΔS in the 3D deformation test as a function of particle number density N_{pc} with grid resolutions of $N_x = 16$, $N_x = 32$, and $N_x = 64$. The dashed line shows first-order convergence and the dotted line shows convergence with the square root.

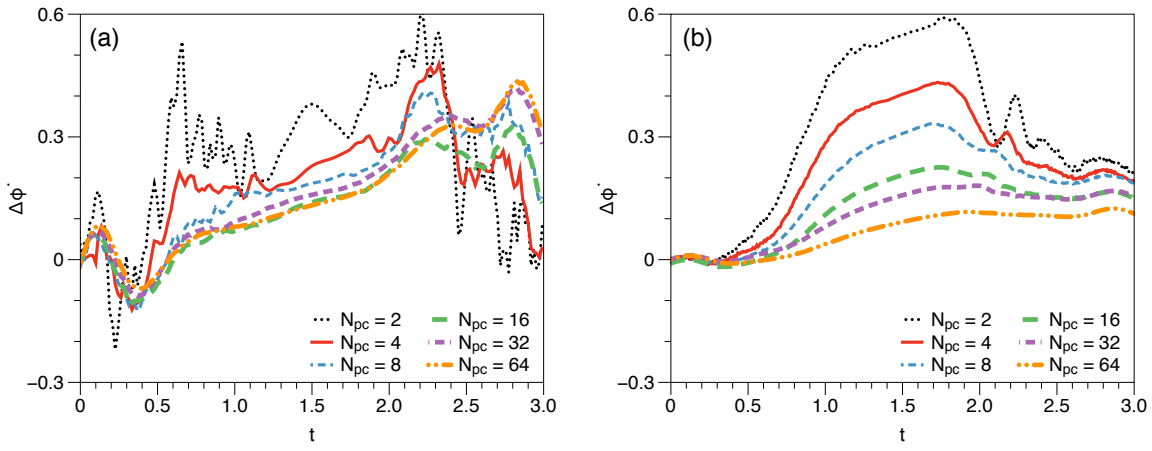

Figure 4.18. Transient percent error in CF conservation in the 3D deformation test as a function of the number of particles per cell N_{pc} at grid resolutions of (a) $N_x = 16$ and (b) $N_x = 64$.

Convergence of shape error is shown in Fig. 4.17. For all three grid resolutions, the shape error decreases at slightly slower than first-order with respect to increasing particle number density over the range of particle number densities considered.

Temporal evolution of the CF conservation error is shown in Fig. 4.18. Results for $N_x = 16$ are shown in Fig. 4.18 panel *a* and results for $N_x = 64$ are shown in Fig. 4.18 panel *b*. A maximum error of approximately 0.6% occurs at a particle number density of $N_{pc} = 2$ for both mesh resolutions. Increasing the particle number density does not decrease the conservation error for the coarse resolution simulations ($N_x = 16$). However, at the moderate resolution of $N_x = 64$, increasing the particle number density from $N_{pc} = 2$ to $N_{pc} = 64$ decreases the error from 0.6% to 0.1%.

Figure 4.19 shows the convergence behavior of the maximum conservation error. The $N_x = 16$ errors do not become smaller with increasing N_{pc}. The $N_x = 32$ and $N_x = 64$ errors monotonically decrease with the square root of N_{pc} over the range of N_{pc} considered.

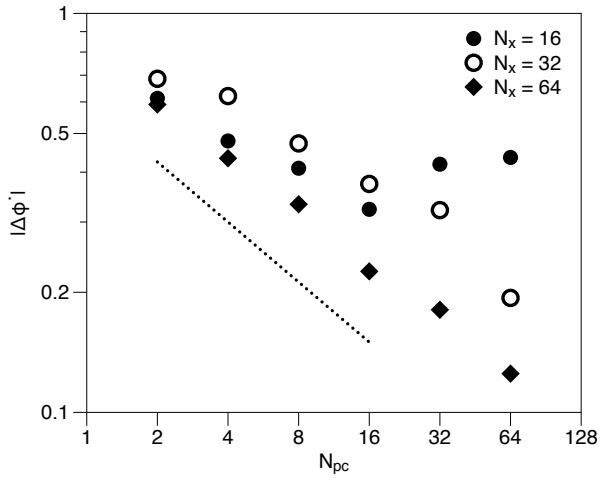

Figure 4.19. Magnitude of maximum percent error $|\Delta\phi^\star|$ in CF conservation in the 3D deformation test as a function of particle number density N_{pc} for grid resolutions of $N_x = 16$, $N_x = 32$, and $N_x = 64$. The dotted line shows convergence with the square root.

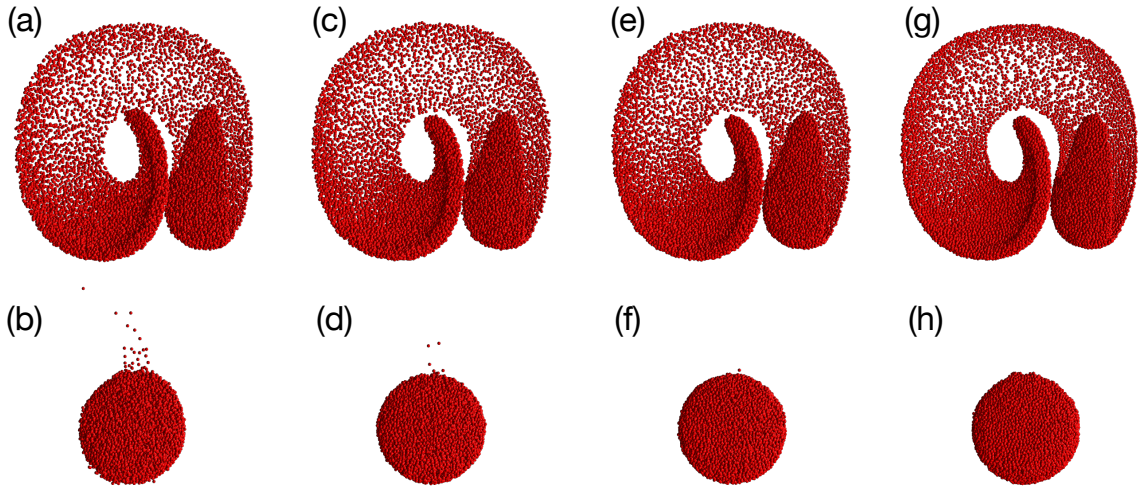

Figure 4.20. Particle distributions for the 3D deformation test at Courant numbers of $C = 12.2$ (panels *a*–*b*), $C = 3.2$ (panels *c*–*d*), $C = 1.6$ (panels *e*–*f*), and $C = 0.24$ (panels *g*–*h*). Results are shown at times $t = 1.5$ (top row) and $t = 3$ (bottom row) with grid resolutions of $N_x = 64$ nodes and particle number densities of $N_{pc} = 4$ particles per cell

Courant number performance

As a final evaluation of phase transport, we consider the dependency of performance on the Courant number, defined here as

$$C = \max\left(\frac{|u|\,dt}{\Delta x} + \frac{|v|\,dt}{\Delta y} + \frac{|w|\,dt}{\Delta z}\right)\Bigg|_{t=0}. \tag{4.42}$$

This evaluation is performed for the three dimensional deformation problem using a setup identical to the previous section. The only difference is that the time step is varied in order to produce a wide range of Courant numbers.

We perform simulations at four Courant numbers: $C = 12.2$, $C = 3.2$, $C = 1.6$, and $C = 0.24$. Each simulation is performed on a grid with resolution $N_x = 64$ and a particle number density of $N_{pc} = 4$ particles per cell. Figure 4.20 shows the particle fields produced by the four Courant numbers. Each Courant number produces a stable result with similar particle distribution at the intermediate time $t = 1.5$. The $C = 12.2$ case in panel *b* includes significant dispersion and shape

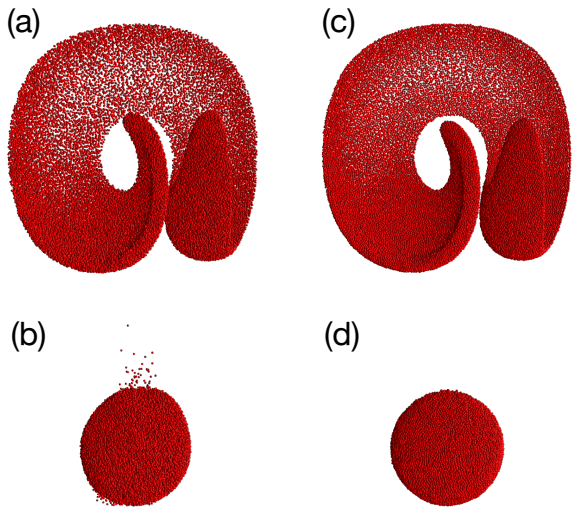

Figure 4.21. Particle distributions for the 3D deformation test at Courant numbers of $C = 12.2$ (panels *a*–*b*) and $C = 0.24$ (panels *c*–*d*). Results are shown at times $t = 1.5$ (top row) and $t = 3$ (bottom row) with grid resolutions of $N_x = 64$ nodes and particle number densities of $N_{pc} = 16$ particles per cell.

errors originating from inaccuracies in the time integration. As the Courant number decreases to $C = 3.2$ and $C = 1.6$, the influence of temporal integration errors decrease and the final geometry becomes more spherical with fewer detached particles. As the Courant number decreases further to $C = 0.24$, small surface anomalies appear that do not arise at Courant numbers closer to $C = 1$. These anomalies result from the residual particle acceleration terms becoming significant relative to the interpolated velocity $\overline{\mathbf{u}}_i$.

Simulations at the extreme Courant numbers of $C = 12.2$ and $C = 0.24$ are performed with an increased particle number density of $N_{pc} = 16$. Figure 4.21 shows the results for these increased particle number density simulations. Panels *a* - *b* show results for $C = 12.2$ and panels *c* - *d* show results for $C = 0.24$. The particle distribution for $C = 12.2$ in panel *a* appears well-behaved, but panel *b* shows detached particles and shape error in the final result. Increasing the particle number density to $N_{pc} = 16$ does not improve performance relative to $N_{pc} = 4$ for $C = 12.2$ because

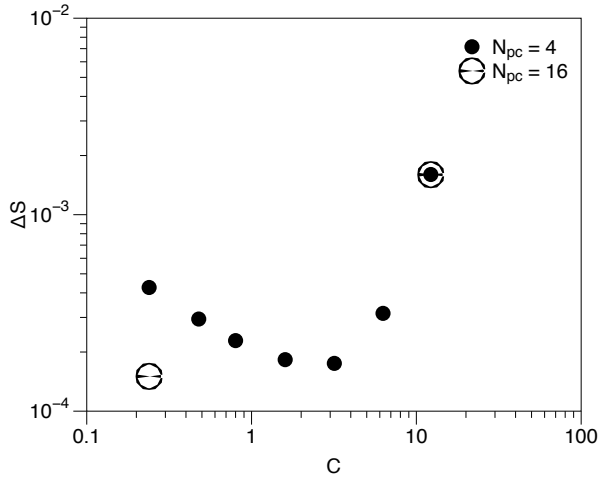

Figure 4.22. Shape error ΔS in the 3D deformation test as a function of Courant number at a grid resolution of $N_x = 64$ nodes and particle number densities of $N_{pc} = 4$ and $N_{pc} = 16$ particles per cell.

the temporal integration produces the errors. The well-distributed particles in panel c and the well-defined sphere in panel d show that increasing the particle number density to $N_{pc} = 16$ eliminates the surface anomalies observed at $C = 0.24$ with $N_{pc} = 4$. At low Courant numbers, increasing the particle number density decreases errors because the length scales associated with the residual particle accelerations become small relative to the interpolated velocity.

Figure 4.22 shows the dependence of shape error on the Courant number. A wide range of results are shown for $N_{pc} = 4$ and results at the extreme Courant numbers are shown for $N_{pc} = 16$. Errors in the time integration dominate at large Courant numbers, with similar errors produced at both $N_{pc} = 4$ and $N_{pc} = 16$. The time integration errors decrease as the Courant number decreases, but errors due to the particle terms increase. For $N_{pc} = 4$ and $N_x = 64$, an optimum Courant number occurs at approximately $C = 3$. At low Courant numbers, increasing the particle number density from $N_{pc} = 4$ to $N_{pc} = 16$ decreases shape error.

4.4.1.5 Comparison to existing methods

Shape error

We compare the PMP shape error in the three-dimensional vortex problem to a number of state-of-the-art volume of fluid methods. The first comparison is made to schemes that transport the VOF at the same resolution as the velocity field (the equivalent, in the context of the PMP, of using one particle per cell $N_{pc} = 1$). For this comparison we consider the unspilt geometric VOF methods of Owkes and Desjardins [104] and Hernández et al. [47]. As a second comparison, we consider the mass-momentum consistent unsplit VOF of Owkes and Desjardins [106], which transports the VOF on a grid with twice the resolution as the velocity (equivalent to eight particles per cell $N_{pc} = 8$).

The PMP method with $N_{pc} = 1$ is compared to the methods of Owkes and Desjardins [104] and Hernández et al. [47] in Fig. 4.23 panel *a*. The PMP method with $N_{pc} = 8$ is compared to Owkes and Desjardins [106] in Fig. 4.23 panel *b*. Panels *a* and *b* show that the PMP method converges with first-order and the three VOF schemes converge with second-order with respect to N_x. The PMP result in panel *a* is more accurate with respect to ΔS than the corresponding VOF methods at resolutions below $N_x = 256$, but less accurate for $N_x = 256$. In panel *b*, the PMP method is more accurate than the corresponding VOF method for all resolutions considered (although the VOF method will out-perform the PMP method for $N_x > 256$).

Conservation error

Many VOF methods conserve mass to machine precision, while the PMP method has a finite conservation error. Although the PMP method does not conserve Eulerian mass, the conservation error is generally low relative to many other non-conservative

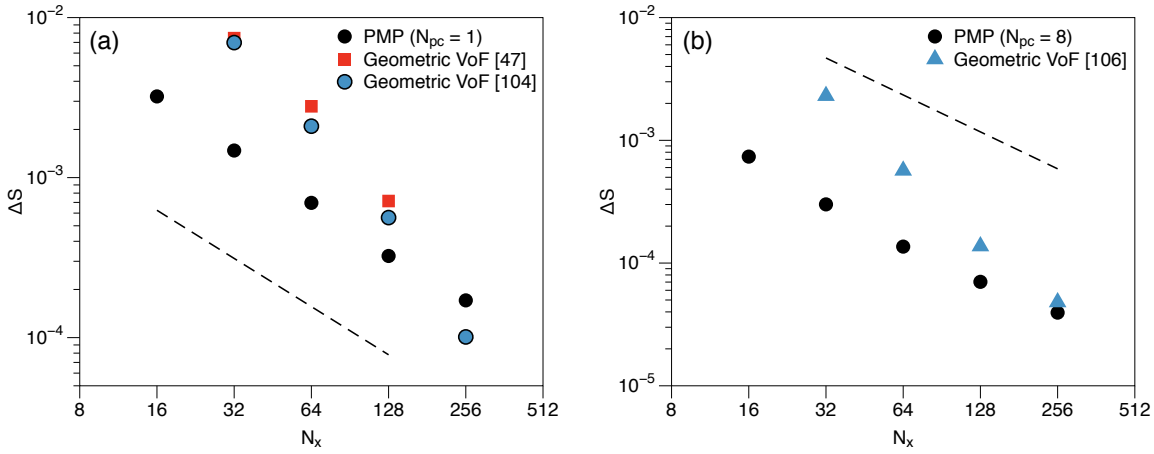

Figure 4.23. Shape error comparison in the three-dimensional vortex problem: (a) PMP method with $N_{pc} = 1$ compared with the geometric VOF methods of Hernández et al. [47] and Owkes and Desjardins [104]; (b) PMP method with $N_{pc} = 8$ compared with the VOF method of Owkes and Desjardins [106]. The dashed line shows first-order convergence.

schemes. For the three-dimensional vortex problem with $N_{pc} = 4$, the PMP conservation error is less than 0.7% for every mesh resolution tested, and less than 0.3% for $N_x \geq 128$. The particle level set method, for example, loses 2.6% of the initial mass for $N_x = 100$ [27] (and the baseline level-set method used in that study suffers an 80% mass loss). In the two-dimensional vortex problem, the PMP method presents slightly less than 0.3% conservation error with $N_x = 128$ and $N_{pc} = 4$. The Refined Local Surface Grid (RLSG) method presents a 0.28% volume loss when using a flow solver grid of $N_x = 128$ and a refined surface grid of $N_x = 1024$ [48]. In the same problem, the Lagrangian particle level set method suffers a 2% mass loss when using a Lagrangian signed distance function with reinitialization on a $N_x = 1000$ grid, utilizing $56,536$ particles [56].

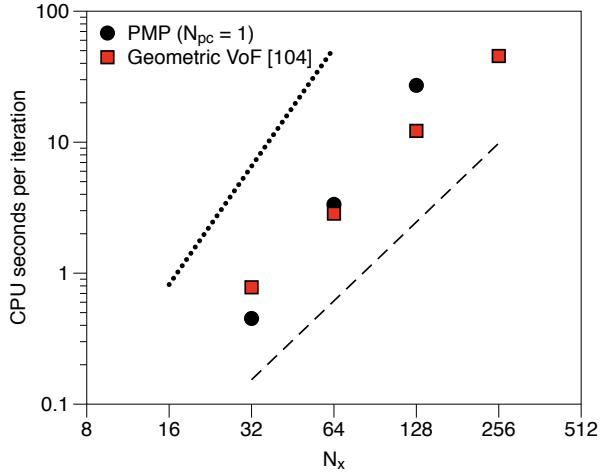

Figure 4.24. Computational cost of performing the three-dimensional vortex problem with the PMP method and the geometric VOF method of Owkes and Desjardins [104]. The dashed line scales with second-order and the dotted line scales with third-order with respect to N_x.

Computational cost

The computational cost of performing the thee-dimensional vortex problem with the PMP method is compared to the unspilt geometric VOF method of Owkes and Desjardins [104] in Fig. 4.24. The PMP simulations were carried out on a 3.3 GHz Intel Core i5 processor with 24 GB 1867 MHz DDR3 memory. The $N_x = 32$ and $N_x = 64$ simulations were run in serial fashion, and the $N_x = 128$ simulation was run on two cores using MPI. The cost of the VOF method scales with surface area, and therefore increases with second order with respect to mesh resolution. The cost of the PMP method, as a discretization of the system mass, scales with third order with respect to mesh resolution. For $N_x \leq 128$, the computational costs of the PMP and the unspilt geometric VOF method [104] are similar. For larger resolutions, the total cost of the VOF method will be significantly less than the PMP method (for a fixed amount of surface area).

The dependence of computational cost on particle number density is shown in

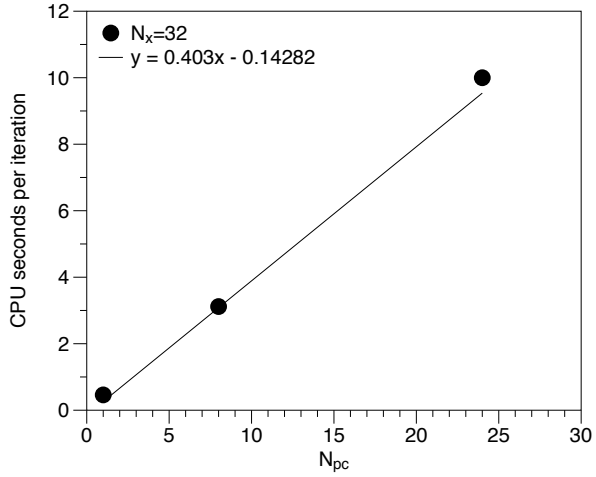

Figure 4.25. Computational cost of performing the three-dimensional vortex problem with the PMP method for $N_x = 32$ as a function of the particle number density N_{pc}.

Fig. 4.25. These results are produced by performing the three-dimensional vortex problem on a mesh with resolution $N_x = 32$. The solid line is linear, and shows the computational cost of the PMP method increases linearly with increasing particle number density N_{pc} – doubling the number of particles doubles the computational cost.

4.4.2 Fluid simulation results

We turn attention to the simulation of multiphase fluid flow problems. Specifically, we demonstrate the solution of the incompressible Navier-Stokes describing the dynamics of an immiscible fluid system with phase-dependent properties.

4.4.2.1 Governing equations

The systems are governed by the momentum equation,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \frac{1}{\rho} \nabla \cdot \left[\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \right], \tag{4.43}$$

where ρ is the density, $\mathbf{u}$ is the velocity vector, P is the pressure, and μ is the viscosity. The divergence-free condition on the velocity field ensures that the flow is incompressible:

$$\nabla \cdot \mathbf{u} = 0. \tag{4.44}$$

The viscosity and density are computed at all locations $\mathbf{x}$ as a function of the CF:

$$\mu(\mathbf{x}) = \sum_{\alpha=1}^{N_\zeta} \mu^\alpha \phi^\alpha(\mathbf{x}), \tag{4.45}$$

and

$$\rho(\mathbf{x}) = \sum_{\alpha=1}^{N_\zeta} \rho^\alpha \phi^\alpha(\mathbf{x}). \tag{4.46}$$

Or, in simplified two-phase form:

$$\mu(\mathbf{x}) = \mu^1 \phi(\mathbf{x}) + \mu^2 (1 - \phi(\mathbf{x})), \tag{4.47}$$

and

$$\rho(\mathbf{x}) = \rho^1 \phi(\mathbf{x}) + \rho^2 (1 - \phi(\mathbf{x})). \tag{4.48}$$

Interfacial dynamics enter the system through the interfacial boundary condition,

$$[P] = \sigma\kappa + 2[\mu]\, \mathbf{n}^T \cdot \nabla\mathbf{u} \cdot \mathbf{n}, \tag{4.49}$$

where $[P]$ is the pressure jump at the interface, σ is the surface tension coefficient, κ is the interfacial curvature, $[\mu]$ is the difference in fluid viscosities, and $\mathbf{n}$ is the interfacial unit normal vector. The surface tension magnitude $\sigma\kappa$ is directly proportional to the curvature κ.

Our strategy for solving the governing equations is largely modeled off of the solver described by Desjardins et al. [21]. The momentum transport equation is solved on a staggered grid with a second-order spatial discretization. The phase and momentum transport are staggered in time and are integrated in time with an iterative Crank–Nicolson scheme as described in Fig. 4.1. The pressure jump at the

interface due to surface tension is imposed with the Ghost Fluid Method (GFM) [32]. The curvature used to define the jump condition for pressure in the GFM is computed with the Variable Finite Particle Method (FPM) described in Chapter 3 and Wenzel and Garrick [149]. A general description of the FPM for computing curvature is provided in Chapter 2 and Wenzel and Garrick [148]. Both density and viscosity are computed from the CF, and are therefore not treated discontinuously with the GFM, but vary over a few mesh spacings.

4.4.2.2 Extending concentric ellipses

We consider extending concentric ellipses, a multiphase problem commonly used to test the performance of SPH methodologies [94, 66, 67]. The notation, general description, and solution of this problem is presented by Monaghan and Rafiee [94], but is repeated here for completeness. The test traditionally involves a circle of radius r_o of phase ζ^1 surrounded by an annulus of phase ζ^2 with inner radius r_o and outer radius R_o. The outer surface of phase ζ^2 is a free-surface, meaning there is no fluid beyond R_o at the initial time t_o. An extensional velocity is imposed at initial time t_o, defined by

$$u(t_o) = \sigma(t_o)x, \tag{4.50}$$

and

$$v(t_o) = -\sigma(t_o)y, \tag{4.51}$$

where σ is a time-dependent scalar. For fluids governed by the Euler equations, the initially concentric circles transition into concentric ellipses defined by

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{4.52}$$

for the interior fluid ζ^1 and

$$\frac{x^2}{A^2} + \frac{y^2}{B^2} = 1 \tag{4.53}$$

for the exterior fluid ζ^2. The aspect ratios increase in time, and the temporal evolution of the major and minor axes of both phases is derived analytically in [94]. A number of relations hold for all times:

$$\frac{A}{B} = \frac{a}{b}, \tag{4.54}$$

$$ab = r_o^2, \tag{4.55}$$

and

$$AB = R_o^2. \tag{4.56}$$

A set of coupled ordinary differential equations defines the rate of change of the major axis:

$$\frac{da}{dt} = \sigma a \tag{4.57}$$

and

$$\frac{d\sigma}{dt} = \sigma^2 \frac{b^2 - a^2}{a^2 + b^2}. \tag{4.58}$$

These equations are solved numerically, and used as a benchmark to assess the performance of multiphase flow simulation methods.

Problem setup

The extending ellipse problem is traditionally performed in the context of Lagrangian methods, where the free-surface does not impose any challenges. The PMP method considered here is a coupled Eulerian-Lagrangian method, and the free-surface introduces challenges associated with boundary conditions. We therefore modify the extending ellipse problem to be appropriate for the PMP methodology, and outline the modifications in this section.

The domain is defined by $x = \{-5.25, 5.25\}$, $y = \{-3.5, 3.5\}$, and $z = \{-1.75, 1.75\}$. The domain is discretized by a uniform mesh of $N_x = 240$, $N_y = 160$, and $N_z = 80$ nodes in the x, y, and z directions, respectively. Simulations are performed with

Table 4.2. Definition of fluid properties for the expanding elliptical regions test problem.

Phase ζ^α	Density ρ^α	Kinematic viscosity ν^α
α=1	5000	0.0001
α=2	1000	0.0001
α=3	1	0.01
α=4	0.1	0.01

particle number densities of $N_{pc} = 1$ and $N_{pc} = 4$. We initialize four phases in the domain as a function of radius $r = \sqrt{x^2 + y^2}$ according to

$$\zeta^\alpha(t_o, r) = \begin{cases} \alpha = 1, & r \leq 1, \\ \alpha = 2, & 1 < r \leq 2, \\ \alpha = 3, & 2 < r \leq 2.5, \\ \alpha = 4, & 2.5 < r. \end{cases} \tag{4.59}$$

The phase-dependent density ρ^α and kinematic viscosity ν^α for each phase ζ^α are defined in Table 4.2. The two inner-most phases, ζ^1 and ζ^2, are high density, low viscosity fluids described by the expanding concentric ellipses problem [94]. The two outer-most phases, ζ^3 and ζ^4, are low density, high viscosity fluids designed to emulate a free-surface. The velocity coefficient σ is initialized as $\sigma(t_o) = 0.4$, and $w(t_o) = 0$.

The extensional flow initialization defined in Eqs. (4.50)-(4.51) works for a free surface (because boundary conditions are not required). In the present study, however, each boundary is treated periodically. To accommodate the periodic boundaries, we initialize a temporary velocity defined by Eqs. (4.50)-(4.51), and then solve a Poisson equation for pressure to project this temporary velocity onto a divergence-free velocity field. Because the densities of the interior phases (ζ^1 and ζ^2) are orders of magnitude greater than the exterior fluids (ζ^3 and ζ^4), the velocities of the interior phases remain well-described by Eqs. (4.50)-(4.51) after the projection, while the velocities of the exterior fluids adjusts to satisfy the periodic boundary conditions.

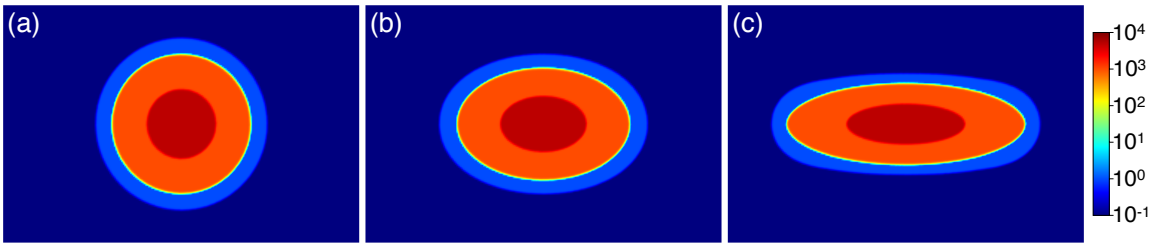

Figure 4.26. Density field in the extending concentric ellipses problem: (a) $t = 0$; (b) $t = 0.5625$; (c) $t = 1.5$. The dark red is ζ^1, the orange is ζ^2, the light blue is ζ^3, and the dark blue is ζ^4. (For interpretation of the colors in the figure, the reader is referred to the web version of this dissertation.)

Results

The temporal evolution of the fluid density is shown in Fig. 4.26 for a particle number density of $N_{pc} = 4$. Panel a shows the initial condition, with ζ^1 as the interior red circle, ζ^2 the orange annulus, ζ^3 the thin, light blue annulus, and ζ^4 the dark blue background fluid. Panel b shows the density distribution at $t = 0.5625$, where the three interior phases have transitioned to ellipses with small aspect ratios. At the final time, $t = 1.5$ in panel c, the ζ^1 and ζ^2 fluids have transitioned to large aspect ratio ellipses with the same aspect ratio. The aspect ratio of the light blue ζ^3 fluid is smaller because it has a larger viscosity. We are interested in quantifying the transition of the two interior phases, ζ^1 and ζ^2, from circular to elliptical geometries.

Figure 4.27 panel a shows the temporal evolution of the major axis of ζ^1, denoted by $2a$. Results are shown for particle number densities of $N_{pc} = 1$ and $N_{pc} = 4$. At the initial time t_o the major axis $2a = 2.0$. As time progresses to the final time of $t = 1.5$, the major axis has increased to $2a \approx 3.5$. For a particle number density of $N_{pc} = 1$, the percent error Δa^* in the major axis $2a$ (defined relative to the solution given by Eqs. (4.50)-(4.51)) is $\Delta a^* = 0.367\%$ after one iteration and $\Delta a^* = 0.817\%$ at the final time of $t = 1.5$. The error roughly doubles over the course of the simulation. For $N_{pc} = 4$, the percent error in the solution is $\Delta a^* = 0.0180\%$ after one iteration and $\Delta a^* = 0.184\%$ at the final time of $t = 1.5$. The error increases by approximately

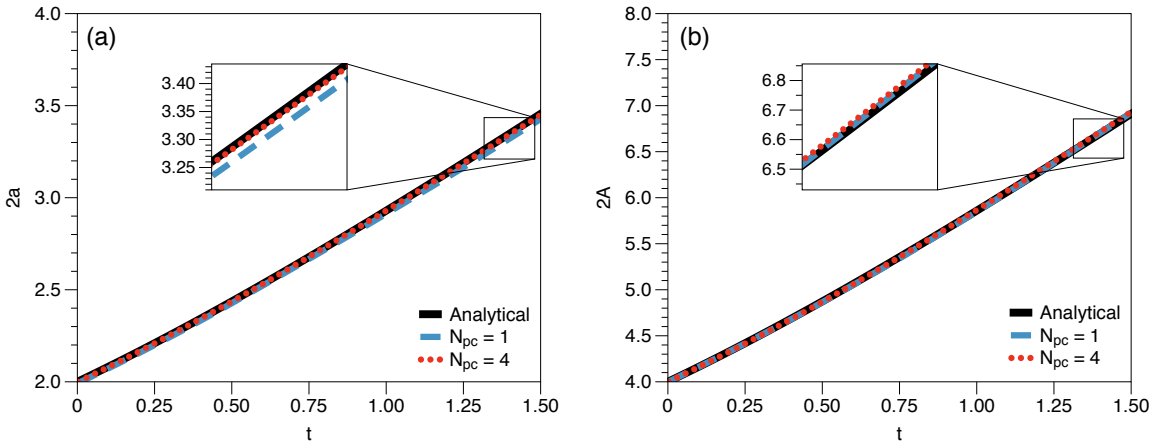

Figure 4.27. Temporal development of the major axis for particle number densities of $N_{pc} = 1$ and $N_{pc} = 4$ compared to the analytical solution: (a) phase ζ^1 axis; (b) phase ζ^2 axis.

a factor of ten. Figure 4.27 panel b shows the temporal evolution of the major axis of ζ^2, denoted by $2A$. For a particle number density of $N_{pc} = 1$, the percent error ΔA^* in the major axis $2A$ is $\Delta A^* = 0.00473\%$ after one iteration and $\Delta A^* = 0.195\%$ at the final time of $t = 1.5$. For $N_{pc} = 4$, the percent error in the solution is $\Delta A^* = 0.0336\%$ after one iteration and $\delta A^* = 0.321\%$ at the final time of $t = 1.5$.

The kinetic energy of the fluid system, comprised of the cumulative kinetic energy of all four phases, is plotted as a function of time in Fig. 4.28. Because the densities of phases ζ^1 and ζ^2 are significantly larger than ζ^3 and ζ^4, the kinetic energy of the system is dominated by the low-viscosity, high-density fluids. The total normalized kinetic energy, defined by $E^\star(t) = E(t)/E(t_o)$, changes by less 0.2% over the duration of the simulations. The kinetic energy does not decay smoothly because the dissipative scales are not resolved in the high density fluids, and the governing equations are solved in non-conservative form. The simulations are stable despite the lack of viscous dissipation and the non-conservative implementation of the momentum equation.

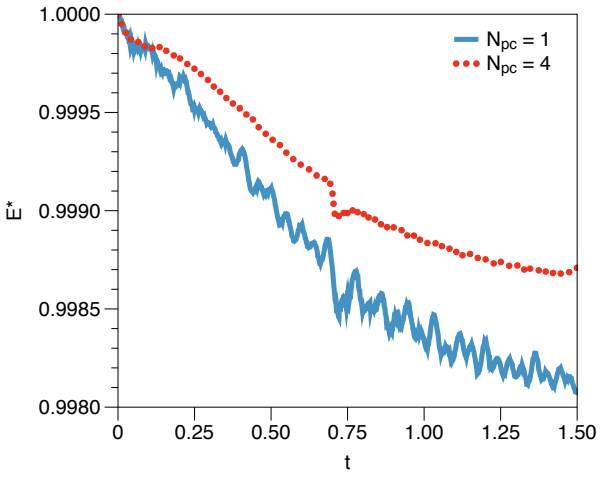

Figure 4.28. Kinetic energy decay for the extending concentric ellipses for particle number densities of $N_{pc} = 1$ and $N_{pc} = 4$.

4.4.2.3 Spurious currents

Spurious currents are unphysical velocities that appear near interfaces in multiphase flow simulations, resulting from errors in computed interfacial curvature. For an initially stationary droplet, the spurious current magnitude is quantified by the spurious capillary number $Ca = |u_{max}|\,\mu/\sigma$. Ideally, the spurious capillary number is zero. We replicate the spurious current assessment performed by Renardy and Renardy [116] and Herrmann [48] to determine the magnitude of spurious currents resulting from coupling the PMP method with the variable FPM method for computing interfacial curvature [149].

Problem setup

The domain is a tri-periodic unit cube discretized by N_x nodes in each direction. We consider mesh resolutions of $N_x = 96$ and $N_x = 128$ and particle number densities of $N_{pc} = 1$ and $N_{pc} = 8$. A sphere of phase ζ^1 with radius $a = 0.125$ is initialized at the center of the domain, which is otherwise filled with phase ζ^2. The density and viscosity of both phases are $\rho = 4$ and $\mu = 1$, the surface tension coefficient is

$\sigma = 0.357$, and the time step $\Delta t = 10^{-5}$. The spurious capillary number is computed after two-hundred iterations.

For the purposes of computing curvature, the color function ϕ is smoothed with a smoothing kernel W^s. We consider two smoothing kernels:

$$W^{s1}(S) = \begin{cases} (1 - S^2)^4, & 0 \leq S \leq 1, \\ 0, & S > 1, \end{cases} \tag{4.60}$$

and

$$W^{s2}(S) = \begin{cases} 1 - S^2, & 0 \leq S \leq 1, \\ 0, & S > 1, \end{cases} \tag{4.61}$$

where $S = |\mathbf{x}_i - \mathbf{x}_j| / h^s$. For all cases, the smoothing length h^s is set to $h^s = 3\Delta x$, where Δx is the Eulerian mesh spacing. A smooth color function ϕ' is then computed according to

$$\phi'(\mathbf{x}_i) = \frac{\sum_{j=1}^{N_j} \phi(\mathbf{x}_j) W^s(\mathbf{x}_i - \mathbf{x}_j, h^s)}{\sum_{j=1}^{N_j} W^s(\mathbf{x}_i - \mathbf{x}_j, h^s)}, \tag{4.62}$$

where N_j is the number of Eulerian cell-centers within distance h^s of $\mathbf{x}_i$ (the cell-center where the smooth color function ϕ' is being computed). The interfacial curvature of the smooth CF ϕ' is computed with the FPM [149, 148].

Results

Table 4.3 provides spurious capillary numbers for the PMP method as a function of mesh resolution and particle number density for smoothing kernel W^{s1}. Every simulation produces a stable result. As a reference for performance, results produced by the continuum surface force (CSF), continuum surface stress (CSS), parabolic reconstruction of surface tension (PROST), and refined local surface grid (RLSG) methods, originally presented by Renardy and Renardy [116] and Herrmann [48], are provided. Like the CSF and CSS methods, the PMP method coupled with the FPM approach for surface tension does not converge with mesh resolution. Results for the

Table 4.3. Spurious capillary numbers for the PMP method with smoothing kernel W^{s1} as a function of mesh resolution N_x and particle number density N_{pc}, compared to archival methods presented by Renardy and Renardy [116] and Herrmann [48].

	$N_x = 96$	$N_x = 128$
PMP, $N_{pc} = 1$	0.0283	0.0348
PMP, $N_{pc} = 8$	0.00843	0.008
CSF [116]	0.00504	0.00516
CSS [116]	0.0106	0.0101
PROST [116]	0.0000628	0.0000367
RLSG [48]	0.0000482	0.0000344

PMP method with $N_{pc} = 1$ are less accurate (larger spurious capillary number) than the CSS and CSF methods. With $N_{pc} = 8$, the PMP method produces a spurious capillary number comparable to those of the CSS and CSF methods. The PROST and RLSG methods are both convergent with respect to spurious capillary number, and greatly out-perform the PMP, CSS, and CSF methods.

Spurious currents are known to depend on interface smoothing procedures [116]. We therefore consider the second smoothing kernel W^{s2} with a mesh resolution of $N_x = 96$. Using the W^{s2} kernel to produce ϕ', rather than W^{s1}, reduces the spurious capillary number by 26% to $Ca = 0.021$ for $N_{pc} = 1$ and by 25% to $Ca = 0.00629$ for $N_{pc} = 8$. We find that changing the smoothing kernel does modify the spurious capillary number.

The pressure jump across the interface of a stationary droplet is given by the Laplace pressure, $\Delta P = 2\sigma/a$. The pressure field is shown in Fig. 4.29 after 200 iterations for $N_x = 128$ and $N_{pc} = 8$ with W^{s1}. Panel *a* shows the pressure in the $x - y$ plane centered at $z = 0$. Because surface tension is being implemented with the ghost fluid method, the pressure changes discontinuously across the interface. Panel *a* appropriately shows that the pressure inside the interface is high, and the pressure outside the interface is low. The percent error in the simulated pressure, computed relative to the pressure jump defined by the Laplace pressure, is shown in

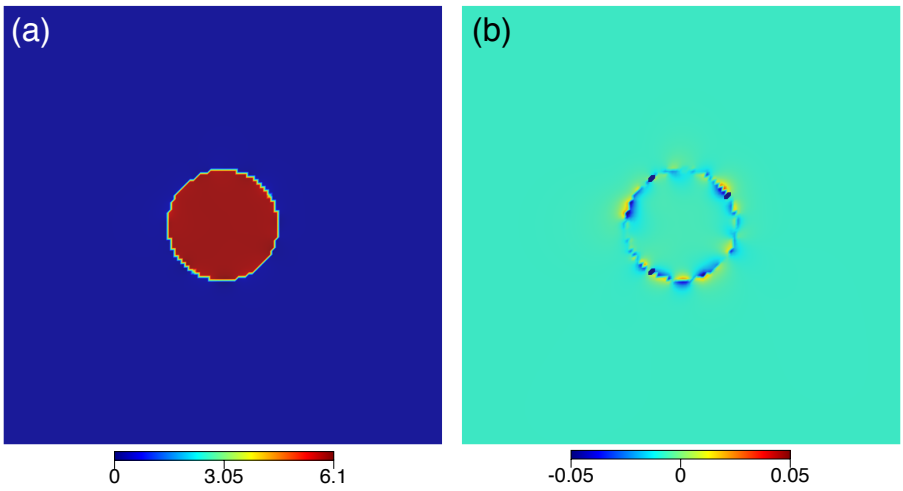

Figure 4.29. Pressure after 200 iterations in the $x-y$ plane centered at $z=0$ in the spurious currents test: (a) Laplace pressure arising from surface tension; (b) percent error in simulated pressure relative to analytical Laplace pressure.

panel *b*. The pressure error is much less than 5% a few mesh spacings away from the interface. Within a few mesh spacings of the interface, the error approaches 5%. At a few nodes in the domain (the three dark blue dots in panel *b*), the numerical interface has crossed over the analytical interface and the error approaches 100%. This behavior is an expected result of the ghost fluid method (a first-order method), where the pressure is treated in discontinuous fashion.

4.4.2.4 Oscillating droplet

We consider the oscillation of an initially elliptical droplet. The problem setup is based on the tests performed by Owkes et al. [107]. The oscillating droplet is a valuable test configuration because it includes phase-dependent properties and surface tension, and the problem dynamics are dominated by the interface. The oscillation has a known period which allows for a quantitative assessment.

Problem setup

The interface between the elliptical droplet and the surrounding fluid is defined at the initial time t_o by

$$\frac{x^2}{(0.1)^2} + \frac{y^2}{(0.12)^2} = 1, \tag{4.63}$$

where the origin is centered on the droplet and the domain extent is $x = \{-.25, .25\}$ and $y = \{-.25, .25\}$. Particles inside the ellipse at t_o are defined by $\chi_i = 1$ and particles outside the ellipse at t_o are defined by $\chi_i = 0$. The droplet has a density of $\rho^1 = 1000$ and a viscosity of $\mu^1 = 8.9 \times 10^{-2}$, and the surrounding fluid has a density of $\rho^2 = 1.3$ and a viscosity of $\mu^2 = 1.81 \times 10^{-3}$. The surface tension coefficient is $\sigma = 0.0728$.

Numerical specification

Grid resolutions of $N_x = N_y = 32$, $N_x = N_y = 64$, and $N_x = N_y = 128$ are considered along with particle number densities of $N_{pc} = 2$, $N_{pc} = 4$, and $N_{pc} = 8$ particles per Eulerian cell. Each simulation domain has a thickness of three nodes in the z direction, similar to the single vortex phase transport test cases. All boundaries are treated periodically.

Potential energy stored in the elliptical interface produces temporal shape oscillations of the droplet. The kinetic energy provides a natural means to examine the oscillation behavior, as the problem is characterized by an exchange between kinetic and potential energy. This work considers the dimensionless droplet kinetic energy $E^\star$,

$$E^\star = \frac{1}{E_o} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2} \rho^1 \phi \mathbf{u} \cdot \mathbf{u} dx dy, \tag{4.64}$$

where E_o is a characteristic energy (defined as the maximum energy observed among all cases). Time is made dimensionless by the analytical period of oscillation [107]

$$t^\star = t \left(2\pi \sqrt{\frac{(\rho^1 + \rho^2)\, a^3}{6\sigma}} \right)^{-1}, \tag{4.65}$$

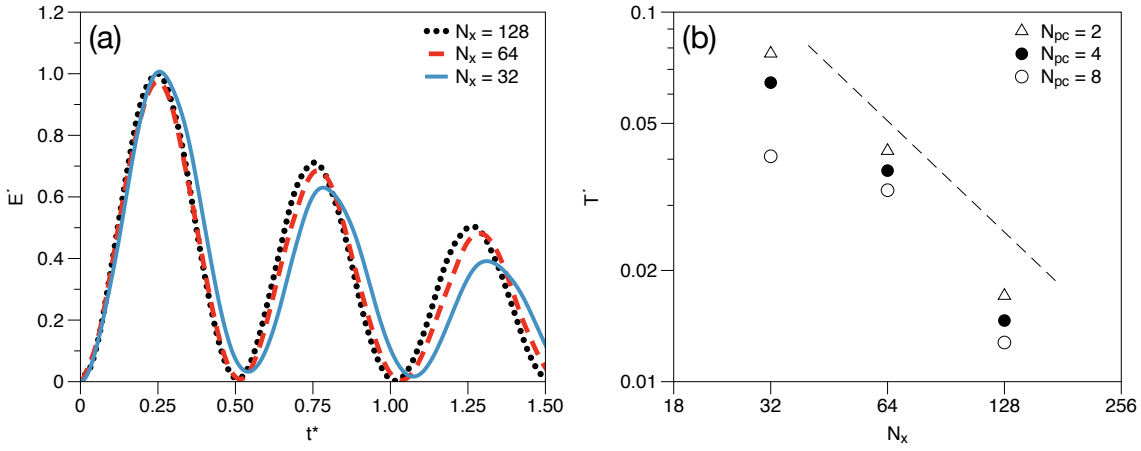

Figure 4.30. Droplet oscillation performance: (a) dimensionless kinetic energy $E^\star$ as a function of time for grid resolutions of $N_x = 32$, 64, and 128 nodes with $N_{pc} = 4$ particles per cell; (b) dimensionless period error $T^\star$ shown for grid resolutions of $N_x = 32$, 48, and 128 nodes and particle number densities of $N_{pc} = 2$, 4, and 8 particles per cell. The dashed line shows first-order convergence.

where $a = \sqrt{0.1 \times 0.12}$ is the undeformed radius of the droplet. The expected behavior of $E^\star$ for an inviscid droplet is a smooth oscillation between $E^\star \approx 0$ and $E^\star \approx 1$ with a dimensionless oscillation period of unity. For a viscous droplet, the amplitude of $E^\star$ will decrease in time due to viscous dissipation. Deviation from the expected oscillation period is defined as the dimensionless period error,

$$T^\star = \mathcal{T} - 1, \tag{4.66}$$

where $\mathcal{T}$ is the measured dimensionless oscillation period, computed by considering the first three local minimums in the dimensionless kinetic energy.

Oscillation frequency

The droplet oscillation period and the error are shown in Fig. 4.30. Panel *a* shows the time-dependent dimensionless kinetic energy for the three $N_{pc} = 4$ cases, and panel *b* shows the dimensionless period error for all nine simulations. All tested grid resolutions produce smooth kinetic energy oscillations, and all tested combinations

of grid resolution and particle number density produce period errors less than 10%. At the highest grid resolution of $N_x = 128$, all three tested particle number densities produced period errors less than 2%. Additionally, as the grid resolution increases from $N_x = 32$ to $N_x = 128$, the oscillation period approaches the expected value of $\mathcal{T} = 1$. Figure 4.30 panel *b* reveals that increasing either the grid resolution, N_x, or the particle number density, N_{pc}, decreases the oscillation period error $T^\star$.

4.4.2.5 Droplet in mean shear

Simulation of a droplet in shear flow is an established means of assessing accuracy against physically-observed results. Droplets exposed to a mean shear either assume a steady deformation or undergo breakup [127, 77]. The behavior depends on the magnitude of the capillary number $Ca = a\dot{\gamma}\mu/\sigma$, where a is the droplet radius and $\dot{\gamma}$ is the imposed shear rate. This problem has been studied extensively by Li et al. [77], where critical capillary numbers have been presented for a wide variety of configurations. We simulate one of the configurations to demonstrate the ability of the point mass formalism to achieve a steady-state deformation under the counteracting effects of viscous shear and surface tension, and to compare the resultant interface to that presented by Li et al. [77].

Problem setup

The geometric setup for the droplet in shear test problem is shown in Fig. 4.31. A droplet of radius a is placed in the center of a domain of size $8a \times 8a \times 4a$ in the x (stream-wise), y (wall-normal), and z (cross-stream) directions, respectively. The grey surfaces at the top and bottom of the domain are solid walls and all other surfaces are periodic. The upper surface moves to the right ($+x$) with speed U_W and the lower surface moves to the left ($-x$) with speed U_W. The initially stationary fluid develops a mean shear rate of $\dot{\gamma} = U_W/4a$ due to boundary layer growth on the solid walls.

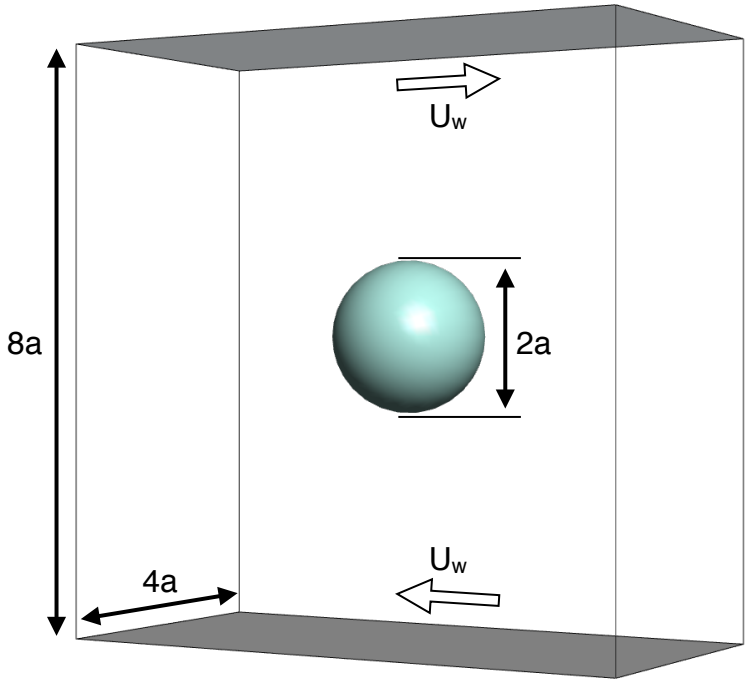

Figure 4.31. Geometric setup for the droplet in shear flow test problem.

In addition to the capillary number, the dynamics depend on the Reynolds number, $Re = \rho\dot{\gamma}a^2/\mu$. The simulation in question is defined by $Ca = 0.1$ and $Re = 20$. Replicating the resolution used in [77], the domain is discretized by $N_x = 64 \times 64 \times 32$ nodes in the x, y, and z directions.

Results

We replicate the $Re = 20$, $Ca = 0.1$ simulation performed by Li et al. [77] at particle number densities of $N_{pc} = 1$ and $N_{pc} = 6$. Figure 4.32 shows results for the $N_{pc} = 6$ simulation. Panel a shows the initial droplet at $t = 0$ (defined by the iso-surface of $\phi = 0.5$), and panels b and c show the droplet at later times of $t = 19$ and $t = 111$. At $t = 19$, the droplet has elongated slightly and rotated clockwise approximately 45° in the $x - y$ plane. Between $t = 19$ and $t = 111$ the droplet drifts slowly to the left. This is an artifact of the droplet center of mass not aligning exactly with

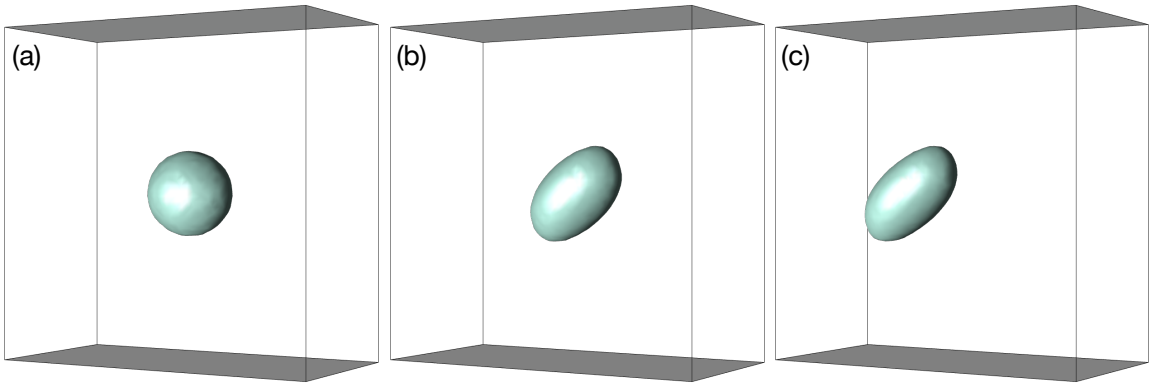

Figure 4.32. Droplet surface (iso surface of $\phi = 0.5$) for the droplet in shear flow test problem with $N_{pc} = 6$ drawn at (a) $t = 0$, (b) $t = 19$, and (c) $t = 111$.

$y = 0$ at $t = 0$. By time $t = 111$, the droplet has achieved a steady ellipsoidal shape that is only slightly different from that at $t = 19$. The steady shape is defined by an equilibrium competition between viscous forces acting to pull the droplet apart and surface tension forces acting to keep the droplet intact.

We assess the droplet in shear results by considering the PMP distribution and CF contour in the $x - y$ plane located at $z = 0$ – the $x - y$ plane that bisects the droplet. Figure 4.33 shows results for the $N_{pc} = 6$ and $N_{pc} = 1$ simulations at $t = 111$. Panels *a* and *b* show the PMP distribution and Eulerian CF for $N_{pc} = 6$. In panel *a*, the particles in both phases are well-distributed, and the interface between the red and blue phases remains well-defined without dispersion. The CF in panel *b* is smooth, well-behaved, and bounded between $\phi = 0$ and $\phi = 1$. Panel *c* shows a comparison between the $\phi = 0.5$ contour (blue dashed line) and the result from Li et al. [77] (thin black line beneath the blue dashed line). The result from Li et al. [77] also includes velocity vectors. The point mass particle solution has been translated in space to compare the droplet shapes. Panel *c* shows good agreement between the steady droplet shapes produced by the point mass technique and the solution presented by [77] – the solid black contour falls beneath the dashed blue contour. Panels *d*, *e*, and *f* are analogous to panels *a*, *b*, and *c*, but correspond to the $N_{pc} = 1$ simulation. Panel

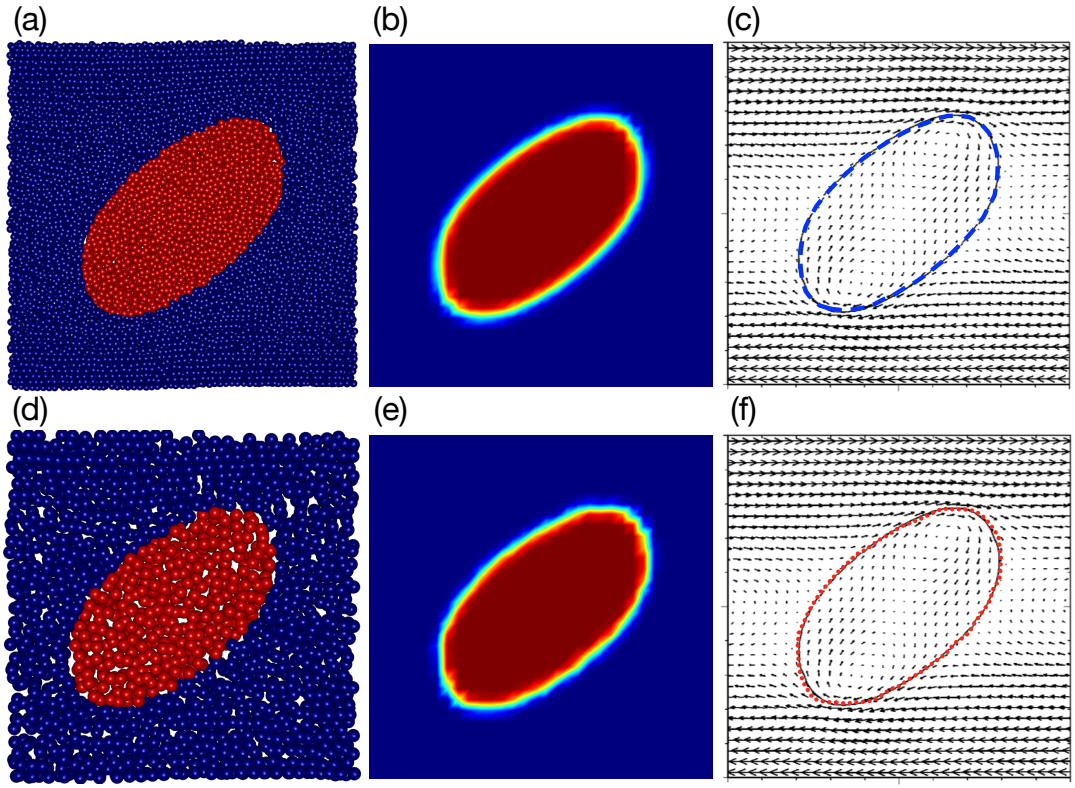

Figure 4.33. Comparisons between point mass particle simulations and the results of Li et al. [77]: (a) particle field for $N_{pc} = 6$ at $t = 111$; (b) Eulerian CF corresponding to the particle field in panel (a); (c) $\phi = 0.5$ contours comparing $N_{pc} = 6$ solution (blue dash) and solution of Li et al. [77] (solid black); (d) particle field for $N_{pc} = 1$ at $t = 111$; (e) Eulerian CF corresponding to the particle field in panel (d); (f) $\phi = 0.5$ contours comparing $N_{pc} = 1$ solution (red dots) and solution of Li et al. [77] (solid black).

d shows well-distributed particles with a well-defined phase interface, despite the low particle number density. The corresponding CF in panel *e* is similarly well-behaved, bounded, and smooth. Panel *f* shows a very slight discrepancy between the $N_{pc} = 1$ solution and the solution from [77].

Capillary breakup

Increasing the dynamic viscosity of the fluid increases the capillary number and decreases the Reynolds number, resulting in capillary breakup according to Li et al.

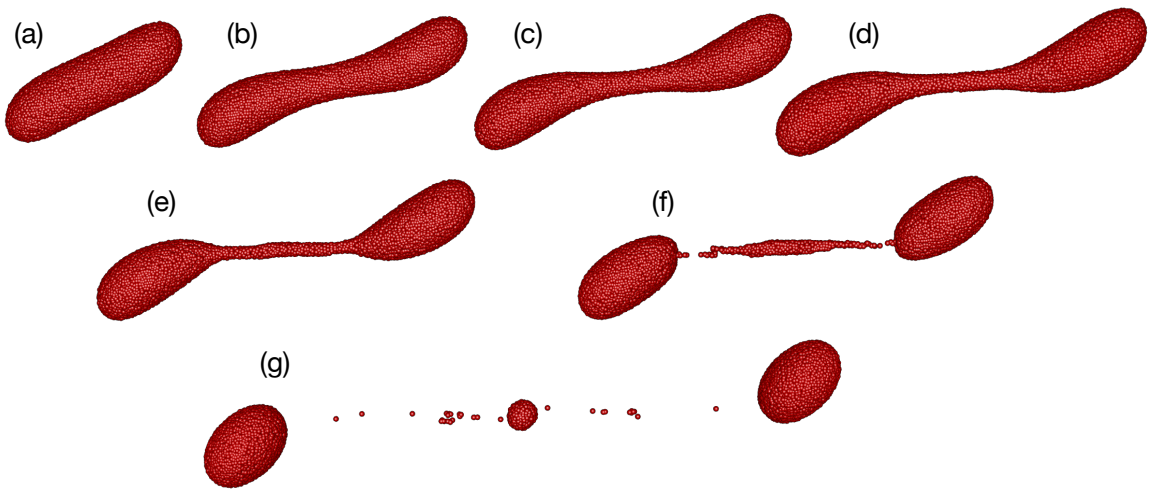

Figure 4.34. Droplet with $Ca = 0.14$ and $Re = 14.3$ undergoing capillary breakup: (a) $t = 64.1$; (b) $t = 74.7$; (c) $t = 76.9$; (d) $t = 77.9$; (e) $t = 79.0$; (f) $t = 80.0$; (g) $t = 84.3$.

[77]. In order to determine if the PMP methodology is capable of undergoing capillary breakup, we increase the dynamic viscosity by 40%, resulting in $Ca = 0.14$ and $Re = 14.3$. To accommodate a longer droplet and the increased range of length scales associated with capillary breakup, we increase the domain size to $16a \times 8a \times 4a$, discretize the domain with a $256 \times 128 \times 64$ Eulerian mesh, and use a particle number density of $N_{pc} = 1$.

Figure 4.34 shows the $Ca = 0.14$ and $Re = 14.3$ droplet undergoing capillary breakup as a function of time. Panel *a* shows the droplet at $t = 64.1$, when it has elongated past the steady geometry of the $Ca = 0.1$, $Re = 20$ droplet previously considered. As time advances, the droplet continues to elongate, necking down at the center, resulting in a prominent ligament by time $t = 77.9$ in panel *d*. By time $t = 80$ in panel *f*, pinch-off has occurred, producing two distinct droplets, a contiguous ligament in the center, and a number of PMPs that have become dispersed in the background fluid. At time $t = 84.3$ in panel *g*, three distinct droplets are apparent, and more PMPs have become dispersed in the background fluid. This dispersal of

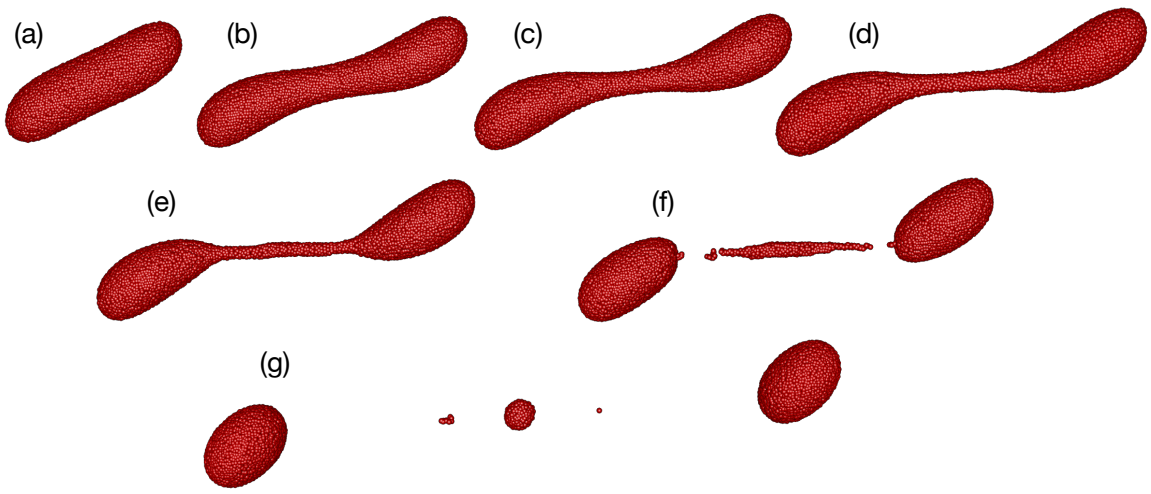

Figure 4.35. Droplet with $Ca = 0.14$ and $Re = 14.3$ undergoing capillary breakup with dispersed particle correction (DPC): (a) $t = 64.1$; (b) $t = 74.7$; (c) $t = 76.9$; (d) $t = 77.9$; (e) $t = 79.0$; (f) $t = 80.0$; (g) $t = 84.3$.

PMPs has occurred because there is little numerical surface tension present in the PMP method, unlike a VOF method.

Implementing a numerical surface tension

A simple technique for adding numerical surface tension to the PMP method is to relocate individual particles that have become surrounded by a different phase. We consider a procedure that, at the end of every time-step, checks for particles that have fewer than four neighboring particles of the same phase within a radius of $1.75\,(V/N_{pc})^{1/3}$, where $(V/N_{pc})^{1/3}$ is the characteristic inter-particle spacing. If a particle i has fewer than four neighboring particles of the same phase, particle i is moved to a new position $\mathbf{X}_i'$ according to

$$\mathbf{X}_i' = \mathbf{X}_i + 1.25 \times (\mathbf{X}_j - \mathbf{X}_i)\,, \tag{4.67}$$

where $\mathbf{X}_j$ is the position of the nearest particle j satisfying $\chi_j^\alpha = \chi_i^\alpha$. Particle i effectively "leap frogs" particle j when moving from position $\mathbf{X}_i$ to $\mathbf{X}_i'$, ideally assuming

a position surrounded by particles of its own phase. The particle motion defined by Eq. (4.67) is referred to as dispersed particle correction (DPC), because it attempts to correct the position of particles that have become dispersed in a dissimilar phase.

Figure 4.35 shows the $Ca = 0.14$ and $Re = 14.3$ droplet undergoing capillary breakup as a function of time with the DPC implementation. The results of this simulation are identical to those of Fig. 4.34 up to the point where the first particle becomes dispersed in the background fluid, which occurs between times $t = 79.0$ in panel *e* and $t = 80.0$ in panel *f*. The onset of the capillary instability is therefore unmodified by the inclusion of DPC. After breakup has occurred, dispersed particles are moved according to the DPC procedure. The resultant particle distribution at time $t = 84.3$ in panel *g* includes the three large droplets present in Fig. 4.35 and two very small droplets (one comprised of only one PMP). The lack of perfect symmetry is attributable to two primary causes. First, perfect symmetry is not a condition imposed on the initial PMP distribution. Second, small errors in the computed curvature can result in asymmetric breakup, especially when the breakup event is not particularly well-resolved.

4.4.2.6 Rayleigh-Taylor instability

In the next test, we simulate a Rayleigh-Taylor instability. The problem setup is based on the test cases presented by Herrmann [48]. We compare the interface produced by the point mass particle technique to the interface produced by our implementation of the split geometric VOF method of Weymouth and Yue [152]. We additionally assess the affects of mesh resolution and particle number density on the interface.

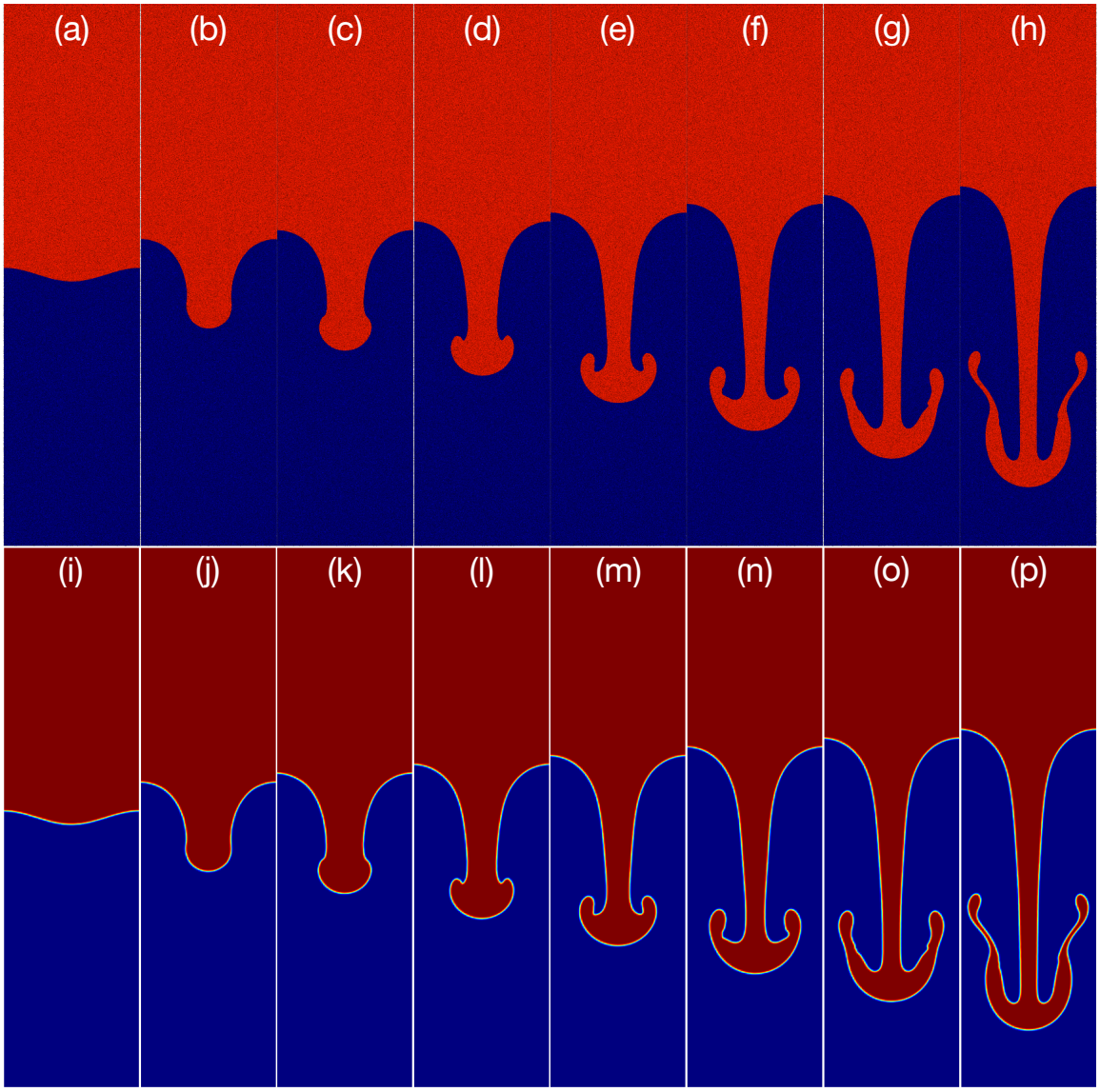

Figure 4.36. Temporal evolution of the particles (top row) and CF (bottom row) in a Rayleigh-Taylor instability on an $N_x = 96$ grid with $N_{pc} = 16$ particles per cell. The times show, increasing from left to right, are $t = 0.0$, $t = 0.5$, $t = 0.6$, $t = 0.7$, $t = 0.8$, $t = 0.9$, $t = 1.0$, and $t = 1.1$.

Problem setup

We perform a Rayleigh-Taylor instability test problem similar to that described by Herrmann [48], but with a larger viscosity. The domain is defined by $x = \{0, 1\}$ and $y = \{-2, 2\}$. We include a thickness of three nodes in the z direction. The upper and lower surfaces are slip walls, and all other surfaces are treated periodically. The mesh is uniform and denoted by the number of nodes in the x direction to remain consistent with prior notation. For example, a $N_x = 64$ mesh includes 64 nodes in the x direction, 256 nodes in the y direction, and 3 nodes in the z direction.

The upper half of the domain is defined as the liquid phase ζ^1 and the lower half of the domain is defined as the gas phase ζ^2. The liquid density is $\rho^1 = 1.225$ and the liquid viscosity is $\mu^1 = 0.00939$, while the gas density is $\rho^2 = 0.1694$ and the gas viscosity is $\mu^2 = 0.00939$. The interface located at $y = 0$ is perturbed at $t = 0$ by a cosine wave with amplitude 0.05. The gravitational constant is set to $g = 9.81\text{m/s}^2$.

Comparison to Geometric VOF

We simulate the Rayleigh-Taylor instability on a mesh of $N_x = 96$ with a particle number density of $N_{pc} = 16$. Figure 4.36 shows the results of the simulation between $t = 0$ and $t = 1.1$. Panels *a*-*h* show the PMP field and panels *i*-*p* show the Eulerian CF. The initial perturbation at $t = 0$ grows slowly until time $t = 0.5$ shown in panels *b* and *j*. Development happens rapidly at later times, and the surface area of the interface grows significantly as gravity pulls the heavy red fluid downward, displacing the lighter blue fluid upward. At the final time of $t = 1.1$ shown in panels *h* and *p*, the interface has developed two long ligaments of red fluid on opposite sides of a central, comparatively thick stem connected to the bulk mass of red fluid at the top of the domain. The PMPs remain well-distributed for all times, and the Eulerian field evolves to reflect the particles with a constant interface thickness.

Using the Rayleigh-Taylor instability problem with $N_x = 96$ and $N_{pc} = 16$, we

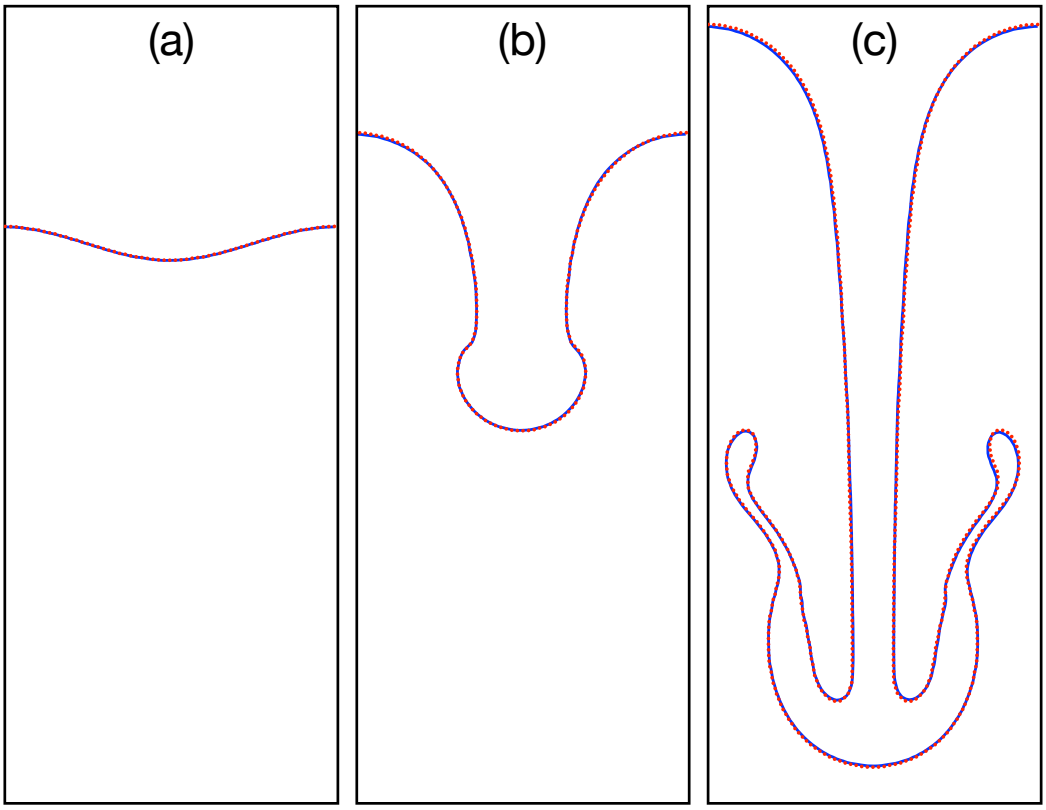

Figure 4.37. Contours of $\phi = 0.5$ on an $N_x = 96$ grid produced by the split, geometric VOF transport scheme of Weymouth and Yue [152] (red dots) and the point mass particle method with $N_{pc} = 16$ (blue lines): (a) $t = 0.0$, (b) $t = 0.6$, and (c) $t = 1.1$.

compare results produced by the PMP method and the geometric VOF method of Weymouth and Yue [152]. Figure 4.37 compares the PMP solution (blue lines) to the geometric VOF solution (red dots). Panel *a* shows the initial condition, panel *b* shows the interface at $t = 0.6$, and panel *c* shows the final solution at $t = 1.1$. In panels *a* and *b*, both solutions are essentially indistinguishable. At time $t = 1.1$, after significant development in the flows, very small differences are visible in the two solutions. All deviations are much smaller than the smallest length dimension present in either solution.

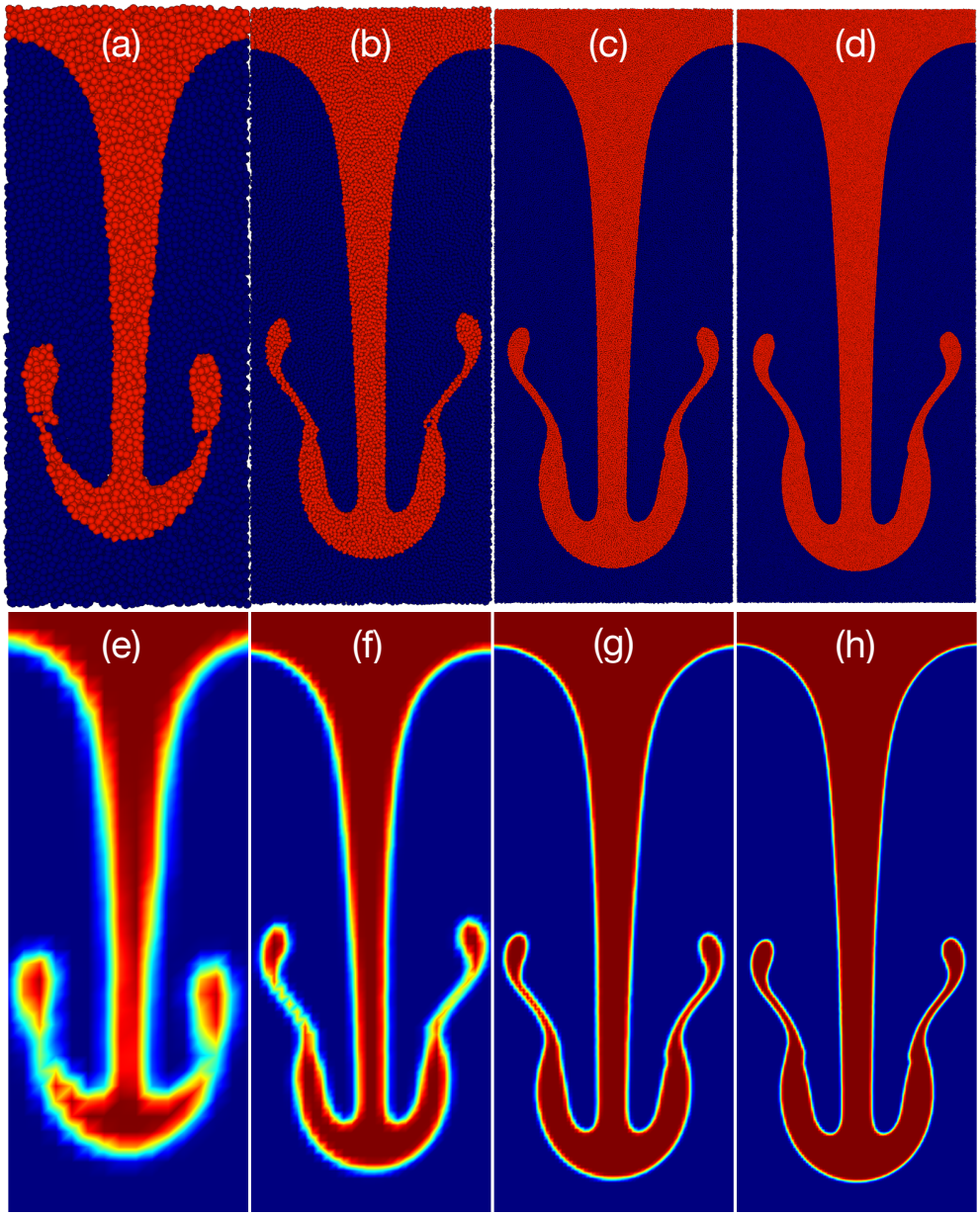

Figure 4.38. Particle fields (top row) and corresponding CF (bottom row) with $N_{pc} = 8$ in the Rayleigh-Taylor instability problem at $t = 1.1$: (a) $N_x = 16$, (b) $N_x = 32$, (c) $N_x = 64$, and (d) $N_x = 96$.

Effects of mesh resolution

We simulate the Rayleigh-Taylor instability with $N_{pc} = 8$ particles per Eulerian cell at four mesh resolutions: $N_x = 16$, $N_x = 32$, $N_x = 64$, and $N_x = 96$. The resulting PMPs and CFs are shown in Fig. 4.38 at $t = 1.1$. The $N_x = 16$ mesh shown in panels *a* and *e* is insufficient to develop the long ligament structures. The ligaments in the red fluid are only resolved by a few particles in panel *a*, and the interfacial thickness on the Eulerian mesh in panel *e* is comparable with the thickness of the ligaments. The penetration depth of the leading edge of the instability is also smaller than the higher resolution cases. At a higher resolution of $N_x = 32$ in panels *b* and *f*, the flow develops the ligament structures and the qualitative shape of the interface, but the penetration depth remains slightly reduced and the flow presents mild asymmetries. The ligaments are well-resolved by the particles, but are comparable with the interface thickness on the Eulerian mesh – they are green, suggesting the feature size is comparable with the interface thickness. As the mesh is further refined to $N_x = 64$ and $N_x = 96$, asymmetries are further reduced and fewer discrepancies are apparent between results.

The total CF $\Sigma\phi$ is computed by integrating the CF over the entire domain. We use the total CF to assess Eulerian conservation behavior in the Rayleigh-Taylor problem. In the case of perfect CF conservation, the analytical value of $\Sigma\phi = 2$ would be constant in time. Percent error in conservation is therefore defined relative to $\Sigma\phi = 2$. Figure 4.39 shows conservation results for the Rayleigh-Taylor problem. Figure 4.39 panel *a* shows the total CF, $\Sigma\phi$, as a function of time, and 4.39 panel *b* shows the maximum percent error. Panel *a* shows that none of the simulations begin with the analytical value of $\Sigma\phi = 2$ because we do not use this as a constraint on the initialization (phase is initialized based on whether a particle is above or below the analytical phase interface, with no other constraints). The time-dependent variation in total CF is small for all cases, but nonetheless decreases rapidly as the mesh resolution increases from $N_x = 16$ to $N_x = 32$. Panel *b* shows that all cases produce

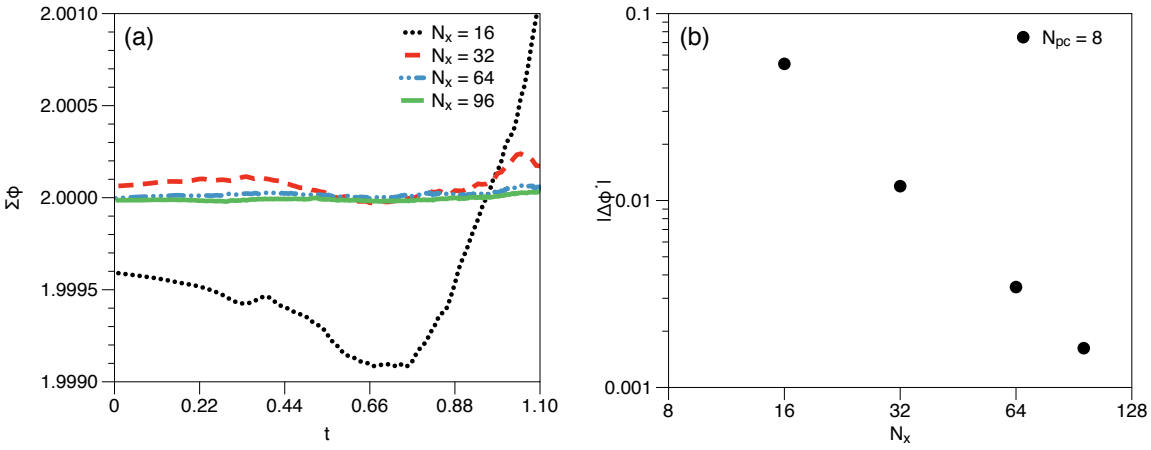

Figure 4.39. Assessment of CF conservation as a function of grid resolution N_x in the Rayleigh-Taylor instability problem with a particle number density of $N_{pc} = 8$ particles per cell: (a) total CF $\Sigma\phi$ as a function of time; (b) magnitude of maximum percent error $|\Delta\phi^\star|$, evaluated relative to the analytical value of total CF, $\Sigma\phi = 2$.

an error of less than 0.06%, with resolutions of $N_x = 64$ and $N_x = 96$ producing errors of less than 0.005%.

Effects of particle number density

We simulate the Rayleigh-Taylor instability on an $N_x = 32$ mesh with three particle number densities: $N_{pc} = 2$, $N_{pc} = 8$, and $N_{pc} = 24$. Results at $t = 1.1$ are shown in Fig. 4.40. Panel *a* shows the solution for $N_{pc} = 2$, panel *b* shows $N_{pc} = 8$, and panel *c* shows $N_{pc} = 24$. The lowest particle number density of $N_{pc} = 2$ generally provides the correct shape, but includes visually obvious asymmetries. As the particle number density increases to $N_{pc} = 24$, these asymmetries are largely eliminated, despite the low mesh resolution of $N_x = 32$.

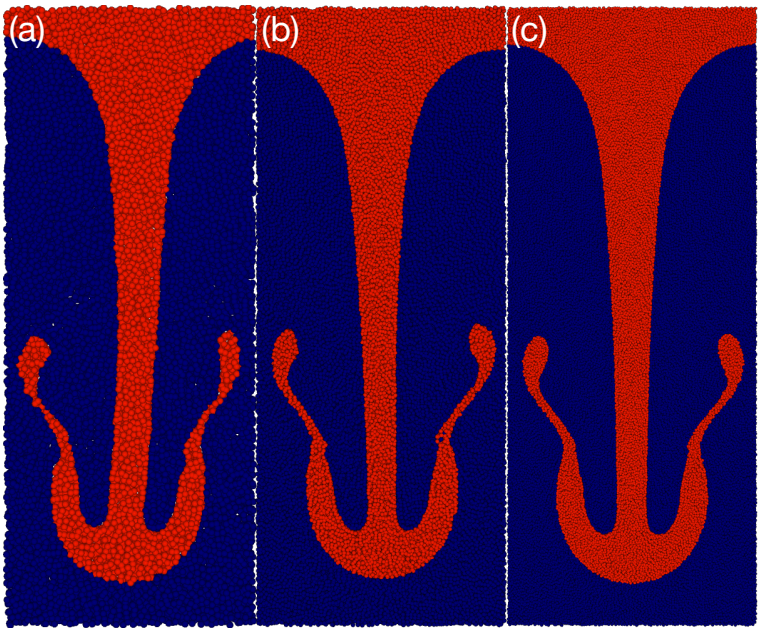

Figure 4.40. Particle fields with $N_x = 32$ in the Rayleigh-Taylor instability problem at $t = 1.1$: (a) $N_{pc} = 2$, (b) $N_{pc} = 8$, and (c) $N_{pc} = 24$.

4.5 Chapter summary and conclusions

This chapter has presented a new Eulerian-Lagrangian approach for the simulation of immiscible multiphase flows. The Navier-Stokes equations are solved on an Eulerian grid while the mass of each phase is discretized and tracked by a set of Lagrangian point mass particles (PMPs). The method is novel in that the motion of the particles is defined by the PMP velocity, which assures the particles remain well-distributed and immiscible. Phase tracking performance is evaluated by means of the canonical single vortex and three-dimensional vortex problems. We illustrate the effects of particle number density, Eulerian grid resolution, Courant number, and model parameters. We also demonstrate the ability of the approach to model an arbitrary number of immiscible fluids. Dynamic performance is assessed via simulation of extending elliptical regions, stationary and oscillating droplets, a droplet in shear flow, and a Rayleigh-Taylor instability.

The methodology exhibits robust performance in the phase transport tests performed. Increasing either the grid resolution or the number of particles eliminates dispersion between phases that occurs at low grid resolutions and particle number densities. Shape errors decrease with first-order and conservation errors decrease with the square root with respect to increasing grid resolution or particle number density. Shape errors are generally smaller than those produced by state-of-the-art, un-split, geometric VOF methods [47, 104, 106] for the grid resolutions tested. Although the PMP method does not conserve mass like many VOF methods, the conservation error is comparable or less than commonly used non-conservative methods [27, 56, 48]. The method successfully transports five phases in the single vortex problem and preserves features below the Eulerian grid resolution. The method is also shown to be stable at all Courant numbers tested ($0.24 \leq C \leq 12.2$). At low Courant numbers, increasing the particle number density decreases transport errors. At high Courant numbers (where error is dominated by the time integration) increasing the particle number density does not decrease transport errors. Similar number density and grid resolution trends are observed in the dynamic problems. The PMP method successfully characterizes the expansion dynamic in the expanding concentric ellipses problem, and presents acceptable spurious current magnitudes and the appropriate pressure jump in stationary droplets with surface tension. In the oscillating droplet simulations, increasing either the particle number density or the grid resolution provides more accurate solutions. (Errors of less than 2% are demonstrated at moderate resolutions and low particle number densities.) Simulations of a droplet in shear flow agree well with previously published studies [127, 77], even with as few as $N_{pc} = 1$ particle per cell. Increasing the viscosity appropriately results in capillary instability, and a scheme is introduced to rectify dispersed PMP particles. In simulations of a Rayleigh-Taylor instability, the PMP methodology produces results that agree with the split geometric VOF method of Weymouth and Yue [152].

The PMP method has a number of novel features and favorable properties. Subgrid-scale phase tracking and superior performance on low-resolution grids are beneficial for the simulation of sprays and other systems where the phase interface is commonly under-resolved. Performance of the PMP method also weakly depends on Courant number, unlike widely used VOF schemes that strictly limit the stable Courant number [152]. Courant number sensitivity is important when an implicit or semi-implicit time integration scheme is used. Additionally, any number of fluids can be considered with the PMP method, which allows for the simulation of general flow problems including many-component mixing. This contrasts with geometric VOF methods, where simulating more than two fluids is a significant challenge due to overlapping volumes [152, 104, 106]. Conversely, some VOF schemes have valuable characteristics that the present PMP methodology does not, including consistent mass-momentum transport and second-order accuracy [106]. Modifying some of the numerical parameters and implementation decisions used in this work can potentially result in similar characteristics without changing the conceptual framework of the PMP method.

Chapter 5

Extension of the point mass particle method to scalar transport with diffusion

5.1 Introduction

Performing scalar transport in the context of interfacial flows is challenging because large gradients develop near interfaces due to rapid changes in fluid properties. Heat transfer presents an additional challenge because specific energy is discontinuous across fluid interfaces. Standard Eulerian convection schemes typically introduce diffusive or dispersive errors when convecting discontinuous quantities. The convection challenge is amplified for interfacial flows, where fluid properties evolve with the flux of VOF, while thermal energy flux is typically evaluated with an entirely different scheme. Using different schemes to compute fluxes of related quantities (density, specific heat, and specific thermal energy) results in inconsistent transport [106]. Lagrangian schemes do not suffer from inconsistent transport because the motion of material elements accounts for convection of all quantities.

This chapter presents and verifies a simple approach for computing scalar transport in the coupled Eulerian-Lagrangian PMP framework introduced in Chapter 4. The motion of the PMPs accounts for convection, which is coupled to diffusion via operator splitting. The approach leverages the SPH diffusion operator of Cleary and Monaghan [16]. The method outlined in this chapter focuses specifically on thermal transport, with an emphasis on robustness in the presence of property discontinuities and large temperature gradients, including boundedness and conservation.

5.2 Formulation

The energy transport equation is expressed in Lagrangian form as

$$\frac{DT}{Dt} = \frac{1}{\rho c}\nabla \cdot \mathrm{k}\nabla T, \tag{5.1}$$

where k is the thermal conductivity, c is the specific heat, ρ is the density, and T is the temperature. Written discretely for particle i, the Lagrangian form becomes

$$\frac{DT_i}{Dt} = \frac{1}{\rho_i c_i}\left[\nabla \cdot \mathrm{k}\nabla T\right]_i, \tag{5.2}$$

where the right-hand-side is the Lagrangian thermal diffusion operator evaluated for particle i. The thermal diffusion operator proposed by Cleary and Monaghan [16] is given by

$$\frac{1}{\rho_i c_i}\left[\nabla \cdot \mathrm{k}\nabla T\right]_i = \frac{1}{\rho_i c_i}\sum_{j=1}^{N_p}\frac{m_j}{\rho_j}\frac{4\mathrm{k}_i\mathrm{k}_j}{\mathrm{k}_i+\mathrm{k}_j}\left(T_i - T_j\right)\frac{\nabla W_{ij}}{\mathbf{X}_i - \mathbf{X}_j}, \tag{5.3}$$

and has been demonstrated to accurately quantify diffusion across interfaces with discontinuous changes in fluid properties, including density, thermal conductivity, and specific heat. Heat diffusion occurs in a pair-wise fashion between particles, ensuring conservation. We propose to modify the diffusion operator of Cleary and Monaghan [16] to be appropriate for use in the context of the PMP method for the solution of convection-diffusion problems.

5.3 Numerical method

5.3.1 Modified diffusion operator

Smoothed particle hydrodynamics simulations compute the density for each particle ρ_i via Eq. (4.4), and this density appears in the diffusion operator given by Eq. (5.3). The PMP approach outlined in Chapter 4 does not require the particle density. This section outlines modifications to Eq. (5.3) that allow diffusion to be computed without computing the particle density. We note that one could, in principle, directly implement Eq. (5.3) to compute diffusion in the context of the PMP method.

Instead of computing the particle density ρ_i on every particle i for use in Eq. (5.3), we assign each particle density (and other fluid properties) according to the particle phase indicator value: if $\chi_i^1 = 1$, then $\rho_i = \rho^1$, $\mathrm{k}_i = \mathrm{k}^1$, and $c_i = c^1$. A consequence of prescribing density as a function of the phase indicator is that the diffusion operator can over-predict or under-predict the diffusion rate between two PMPs if the density assigned to the particle differs significantly from the density defined by Eq. (4.4) (this would happen if the particle field becomes disorganized). Explicitly assigning particle densities according to phase requires modifications to be made to the diffusion operator.

The weight function used in this work is a modified version of the M_4 cubic spline [113]

$$W^q(q) = \frac{1}{\pi h_q^3} \begin{cases} 0, & q < 0.1; \\ \frac{1}{4}(2-q)^3 - (1-q)^3, & 0.1 \le q < 1; \\ \frac{1}{4}(2-q)^3, & 1 \le q < 2; \\ 0, & q \ge 2, \end{cases} \tag{5.4}$$

where $q = |\mathbf{X}_i - \mathbf{X}_j| / h_q$ and $h_q = 1.2\,(V/N_{pc})^{1/3}$. The weight is modified in that $W^q(q) = 0$ for $q < 0.1$, which prevents heat exchange between particles that have

become too close in proximity. Another consequence of prescribing the particle densities, rather than computing them via Eq. (4.4), is that the discrete normalization requirement for the weight function is not satisfied [93]. We therefore introduce a pre-factor to the weight, c_n, resulting in a new weight function that better satisfies the normalization criteria:

$$W^q(q) = \frac{c_n}{\pi h_q^3} \begin{cases} 0, & q < 0.1; \\ \frac{1}{4}(2-q)^3 - (1-q)^3, & 0.1 \leq q < 1; \\ \frac{1}{4}(2-q)^3, & 1 \leq q < 2; \\ 0, & q \geq 2, \end{cases} \tag{5.5}$$

where $q = |\mathbf{X}_i - \mathbf{X}_j| / h_q$ and $h_q = 1.2\,(V/N_{pc})^{1/3}$. The diffusion operator used in this work is now defined by

$$\frac{1}{\rho_i c_i}\left[\nabla \cdot \mathrm{k}\nabla T\right]_i = \mathfrak{D}_i = \frac{1}{\rho_i c_i} \sum_{j=1}^{N_p} \frac{m_j}{\rho_j} \frac{4\mathrm{k}_i \mathrm{k}_j}{\mathrm{k}_i + \mathrm{k}_j} (T_i - T_j) \frac{\nabla W_{ij}^q}{\mathbf{X}_i - \mathbf{X}_j}, \tag{5.6}$$

where the particle density is defined by the particle phase, and the weight W_{ij}^q contains the modifications prescribed in Eq. (5.5). In this work, we select $c_n = 1.13$.

5.3.2 Time integration

A simple Lie-Trotter operator-split approach for time integration of temperature in the context of the PMP method is proposed as follows:

1. The thermal energy of the system is discretized by the PMPs at time level $n - 1/2$, such that the distribution of thermal energy is defined by the particle temperature $T_i^{n-1/2}$, the particle mass M_i, and the particle position $\mathbf{X}_i^{n-1/2}$.

2. The position of each PMP i is advanced from time level $\mathbf{X}_i^{n-1/2}$ to time level $\mathbf{X}_i^{n+1/2}$ via the procedures outlined in Section 4.3.2. This motion accounts for thermal convection.

3. The diffusion operator is computed for each particle i using the most up-to-date particle position $\mathbf{X}_i^{n+1/2}$ and the particle temperature $T_i^{n-1/2}$

$$\mathfrak{D}_i = \frac{1}{\rho_i c_i} \sum_{j=1}^{N_p} \frac{m_j}{\rho_j} \frac{4\mathrm{k}_i \mathrm{k}_j}{\mathrm{k}_i + \mathrm{k}_j} \left(T_i^{n-1/2} - T_j^{n-1/2} \right) \frac{\nabla W_{ij}^{q\ n+1/2}}{\mathbf{X}_i^{n+1/2} - \mathbf{X}_j^{n+1/2}}. \tag{5.7}$$

4. The temperature at time level $n + 1/2$ is computed according to

$$T_i^{n+1/2} = T_i^{n-1/2} + \Delta t \mathfrak{D}_i. \tag{5.8}$$

5.4 Performance assessment

5.4.1 Diffusion in multiphase slabs

As a first test problem we consider diffusion across a phase interface separating water and air. This test is modeled off work presented by Cleary and Monaghan [16].

Problem setup

The problem domain is a rectangle with dimensions 1.5 m $\times$ 0.5 m in the x and y directions, respectively. The mid-point of the domain in the x direction is defined by $x_m = 0.75$ m. The left half of the domain, $x < x_m$, is filled with air, and the right half of the domain, $x \geq x_m$, is filled with water. The air is initially cool with a temperature of $T_l = 300$ K, and the water is initially slightly warmer, with a temperature of $T_r = 301$ K. The air properties are prescribed to be $\rho_l = 1.275$ kg/m^3, $\mathrm{k}_l = 0.02587$ W/m $\cdot$ K, and $c_l = 718$ J/kg $\cdot$ K, and the water properties are prescribed to be $\rho_r = 1000$ kg/m^3, $\mathrm{k}_r = 0.598$ W/m $\cdot$ K, and $c_r = 4181.3$ J/kg $\cdot$ K. The top and bottom of the domain are treated periodically, and the left and right boundaries are adiabatic.

As time advances, the warm liquid phase diffuses thermal energy into the gas phase. The analytical solution to this problem is given by [16]

$$T(x,t) = \begin{cases} T_l + T_c \times \text{Erfc}\left((x_m - x) / \left(2\sqrt{\alpha_l t}\right)\right), & x < x_m; \\ T_l + T_c \times \left[1 + \frac{k_l}{k_r}\sqrt{\frac{\alpha_r}{\alpha_l}}\text{Erf}\left((x - x_m) / \left(2\sqrt{\alpha_r t}\right)\right)\right], & x \geq x_m, \end{cases} \tag{5.9}$$

where

$$T_c = (T_r - T_l)\sqrt{\alpha_l} / \left(\sqrt{\alpha_r} + \sqrt{\alpha_l}\right), \tag{5.10}$$

and $\alpha = k/\rho c$ is the thermal diffusivity.

Numerical parameters

We consider three uniform meshes, defined by $N_x = 30$, $N_x = 60$, and $N_x = 120$ nodes in the x direction (and one-third as many nodes in the y direction). Each mesh includes three nodes in the z direction (much like the tests presented in Chapter 4). For each mesh resolution, we consider a particle number density of $N_{pc} = 1$ particle per Eulerian cell. The time step is defined according to Cleary and Monaghan [16]

$$\Delta t = 0.144 \left(\Delta x\right)^2 / \alpha_l. \tag{5.11}$$

Introducing a vertical velocity $v(x)$ perturbs the relative particle positions, ensuring the particles do not assume an ideal, perfectly-distributed configuration. We therefore introduce a velocity $v(x)$ defined by

$$v\left(x\right) = \frac{2x_m - x}{2x_m}\frac{\Delta x}{\Delta t}. \tag{5.12}$$

Adding this vertical velocity does not change the analytical solution provided by Eq. (5.9). Comparison to the analytical solution is performed via the maximum percent error, defined by

$$L_\infty(t) = \max\left[100 \times \frac{|T_i(t) - T(x,t)|}{\Delta T_o}\right], \tag{5.13}$$

where $\Delta T_o = T_r - T_l$ is the initial temperature difference.

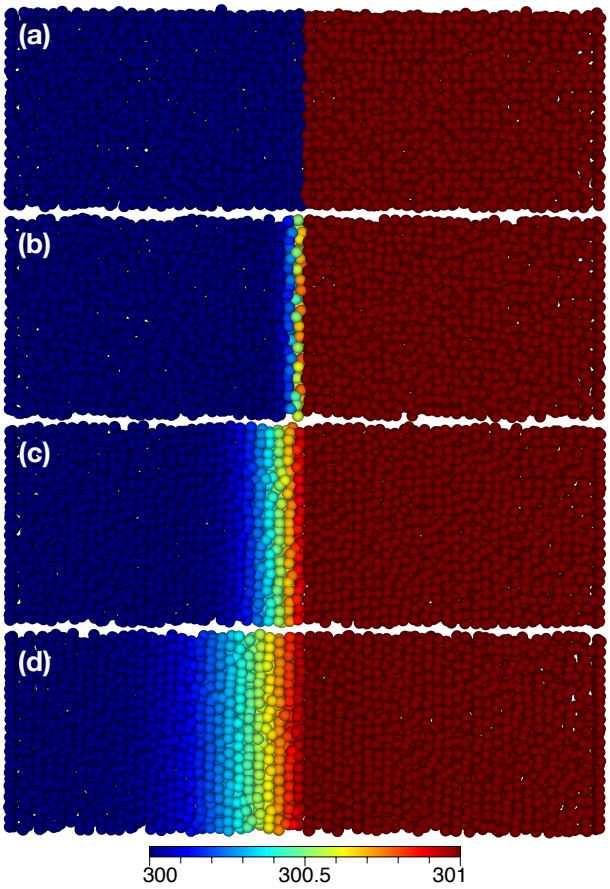

Figure 5.1. Point mass particles colored by temperature (in units of Kelvin) in the multiphase slab diffusion problem for $N_x = 60$: (a) $t = 0$ s; (b) $t = 15.9$ s; (c) $t = 175$ s; (d) $t = 637$ s.

Results

The PMPs from the $N_x = 60$ simulation are shown, colored by temperature, in Fig. 5.1. Panel *a* shows the initial condition, $t = 0$, and panels *b*, *c*, and *d* show the particle field at times $t = 15.9$ s, $t = 175$ s, and $t = 637$ s, respectively. The temperature field is initially discontinuous, with the water warm relative to the air. As time progresses, energy diffuses from the water to the air. The air near the interface visibly elevates in temperature, while the water temperature does not visibly change due to its relatively high heat capacity.

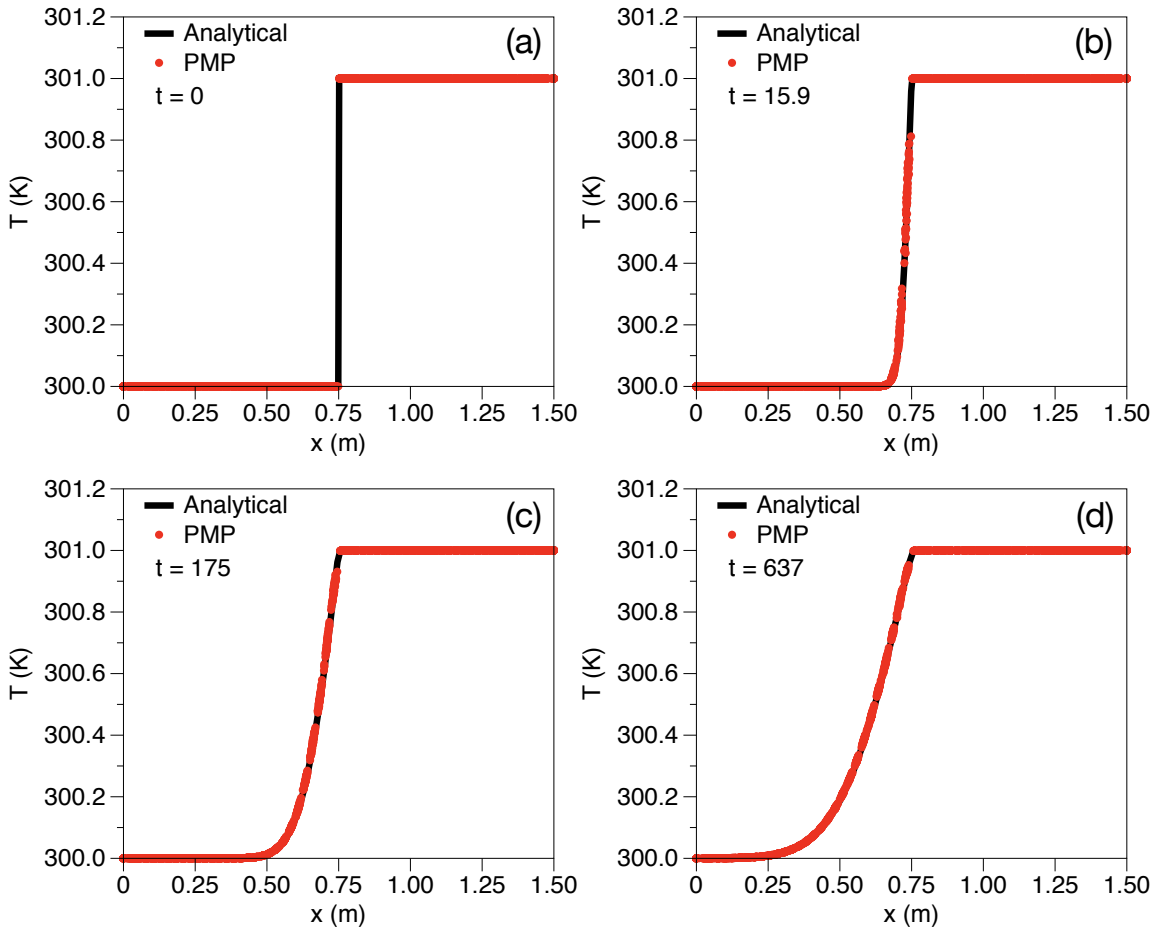

Figure 5.2. Comparison of the PMP temperatures to the analytical temperature in the multiphase slab diffusion problem for $N_x = 60$: (a) $t = 0$ s; (b) $t = 15.9$ s; (c) $t = 175$ s; (d) $t = 637$ s.

Figure 5.2 compares the PMP temperatures for the $N_x = 60$ simulation to the analytical solution provided by Eq. (5.9). Results are shown for the same times as presented in Fig. 5.1. The temperature of every PMP in the domain is plotted as a point, and the analytical solution is shown as a line. Panel *a* shows the initial condition, where the PMP solution agrees identically with the analytical solution. After a short time of $t = 15.9$ in panel *b*, very slight discrepancies appear between the PMP solution and the analytical solution. These discrepancies do not grow as time progresses in panels *c* and *d*.

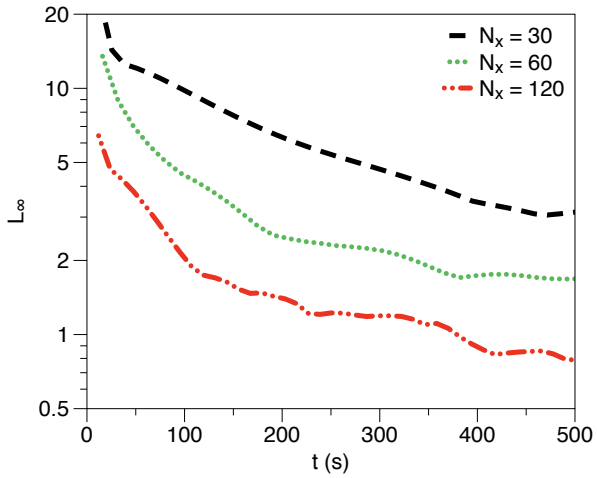

Figure 5.3. Maximum percent error in the multiphase slab diffusion problem as a function of time for mesh resolutions of $N_x = 30$, $N_x = 60$, and $N_x = 120$.

The maximum error in temperature is graphed as a function of time in Fig. 5.3 for $N_x = 30$, $N_x = 60$, and $N_x = 120$. At each resolution, the maximum percent error is initially large because of the initial temperature discontinuity. As time progresses, the maximum error L_∞ decreases for each mesh resolution. Error also decreases with increasing mesh resolution. For a resolution of $N_x = 120$, the final percent error is less than 1%. Each of the simulations conserves thermal energy and preserves boundedness to machine precision.

5.4.2 Convection and diffusion of a Gaussian

As a second test problem, we consider convection and diffusion of a Gaussian in one-dimension. We compare the results to an analytical solution, consider boundedness of the solution, and again verify thermal energy conservation.

Problem setup

The problem domain is a thin rod with a length $L = 1$ m in the x direction. The domain is filled with a single fluid that has a density of $\rho = 1$ kg/m^3 and specific heat

of $c = 1$ J/kg $\cdot$ K. Thermal conductivity is considered at values of k $= 0.002$ W/m $\cdot$ K and k $= 10^{-12}$ W/m $\cdot$ K, serving as high and low diffusivity test cases, respectively. All of the fluid in the domain travels from left to right with a speed of $u_o = 1$ m/s. The temperature is initialized with a Gaussian distribution defined by

$$T(x, t_o) = T_o + A \times \exp\left(-\frac{(x/L - x_o/L)^2}{\sigma^2}\right), \tag{5.14}$$

where $T_o = 300$ K, $A = 50$ K, $x_o = 1/8$ m, and $\sigma = 0.02$. All boundaries are treated periodically to emulate an infinite domain.

As time advances, the initial Gaussian distribution travels from left to right and widens as a function of the thermal diffusivity. The analytical solution to this problem is given by

$$T(x, t) = T_o + \frac{A}{\sqrt{1 + (4\mathrm{k}t/L^2)/\sigma^2}} \exp\left(-\frac{((x/L - u_o t/L) - x_o/L)^2}{\sigma^2 + 4\mathrm{k}t/L^2}\right). \tag{5.15}$$

Numerical parameters

Three mesh resolutions are considered: $N_x = 90$, $N_x = 180$, and $N_x = 360$. The mesh is uniform for each resolution, and in each case there are three mesh spacings in the y and z directions. Each simulation uses a particle number density of $N_{pc} = 1$. For the low diffusivity cases (k $= 10^{-12}$ W/m $\cdot$ K), the time step is defined by

$$\Delta t = \frac{u_o}{\Delta x}, \tag{5.16}$$

and for the high diffusivity cases (k $= 0.002$ W/m $\cdot$ K) the time step is defined by

$$\Delta t = 0.144 \left(\Delta x\right)^2 / \alpha. \tag{5.17}$$

Performance is again measured against the analytical solution via the maximum percent error

$$L_\infty(t) = \max\left[100 \times \frac{|T_i(t) - T(x, t)|}{\Delta T_o}\right], \tag{5.18}$$

where $\Delta T_o = 50$ K is the initial temperature difference.

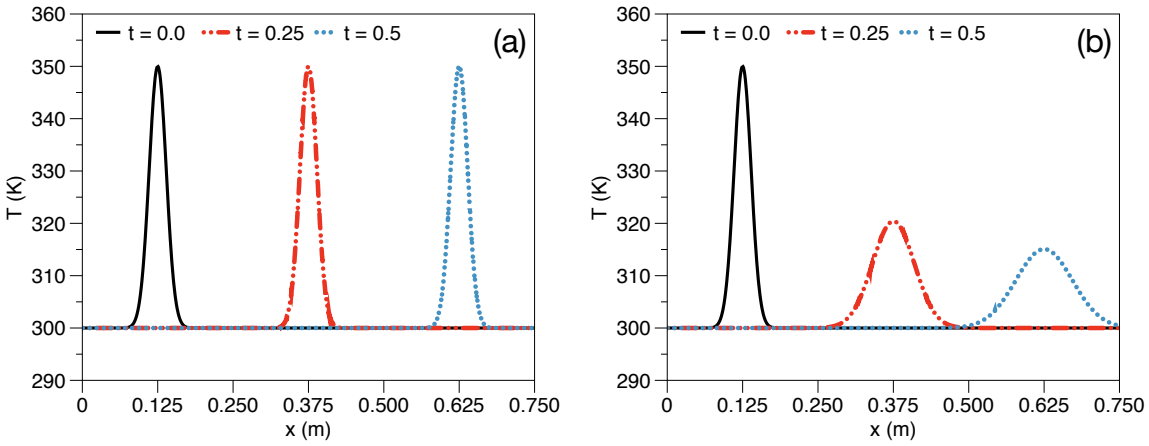

Figure 5.4. Temporal evolution of temperature profiles for convecting and diffusing Gaussians for $N_x = 360$: (a) k = 10^{-12} W/m · K and (b) k = 0.002 W/m · K.

Results

Figure 5.4 shows the PMP solution for a mesh resolution of $N_x = 360$ at times $t = 0$ s, $t = 0.25$ s, and $t = 0.5$ s, with the low diffusivity case in panel *a* and the high diffusivity case in panel *b*. In both panels, the initial condition is shown as the solid line, the solution at $t = 0.25$ s is shown as the dashed-dotted line, and the solution at $t = 0.5$ s is shown as the dotted line. As time progresses, the initial Gaussian profile convects from left to right with velocity u_o. In panel *a*, the Gaussian profile convects from left to right with no visible changes because the diffusion is negligible. In panel *b*, the initial Gaussian travels from left to right with the same convective velocity, but diffusion reduces the maximum value and distributes the thermal energy in space.

Figure 5.5 shows the maximum percent error in temperature as a function of time. Panel *a* shows the low diffusivity results for all three mesh resolutions. Performance is relatively independent of mesh resolution, and in each case the error grows in time. Maximum errors are less than 2%. Panel *b* shows the high diffusivity results. In each case, the error grows rapidly at early times, and then decreases gradually. Increasing the mesh resolution decreases the maximum error, and for each resolution

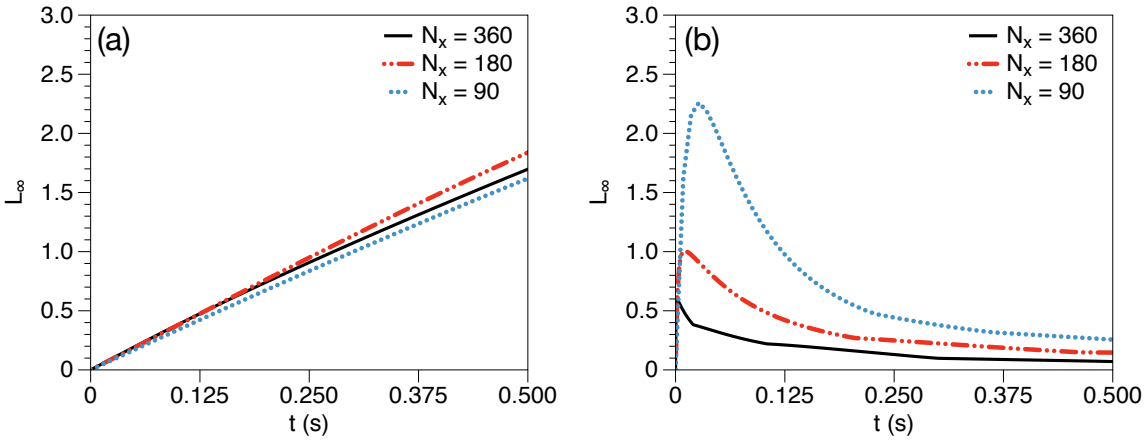

Figure 5.5. Maximum percent error in the convecting and diffusing Gaussians test problem as a function of time for mesh resolutions of $N_x = 90$, $N_x = 180$, and $N_x = 360$: (a) k $= 10^{-12}$ W/m $\cdot$ K and (b) k $= 0.002$ W/m $\cdot$ K.

the final error is less than 0.5%. All six simulations presented in Fig. 5.5 conserve thermal energy to machine precision, and each solution remains bounded by the initial maximum and minimum temperatures.

5.4.3 Air blast atomization with heat transfer

To demonstrate that the PMP method can be used to solve problems of practical interest, we simulate the air blast atomization of a low temperature droplet in hot gas. The purpose of this section is to demonstrate the practical capability of the method, not to produce archival results. We therefore do not provide every detail of the simulation setup.

Problem description

Air blast atomization describes the scenario where fast moving air flows over a low speed or stationary liquid. Breakup requires high dynamic pressure and low viscosity in the gas phase, and depends on the surface tension coefficient, the liquid density,

and the liquid viscosity [57]. The dynamics of single droplet air blast atomization therefore depend on the gas phase Reynolds number $Re = \rho_g D_o U_\infty / \mu_g$, the gas phase (aerodynamic) Weber number $We = \rho_g U_\infty^2 D_o / \sigma$, and the liquid Ohnesorge number $Oh = \mu_L / \sqrt{\rho_L D_o \sigma}$, where subscript g references the gas phase, subscript L references the droplet phase, D_o is the initial droplet diameter, U_∞ is the free-stream velocity, and μ is the dynamic viscosity [63]. At large Reynolds numbers and low Ohnesorge numbers, breakup occurs above a critical Weber number of approximately $We_c \approx 12$ [63]. We therefore consider a flow configuration with $Re = 1341$, $Oh = 0.039$, and $We = 101$. These parameters are achieved by assigning the following gas phase properties

$$\rho_g = 1 \text{ kg/m}^3, \qquad \mu_g = 1.81 \times 10^{-5} \text{ kg/ (m} \cdot \text{s)}, \tag{5.19}$$

and property ratios of

$$\rho_L / \rho_g = 15, \qquad \mu_L / \mu_g = 20. \tag{5.20}$$

The surface tension coefficient is set to $\sigma = 0.006$ N/m and the initial droplet diameter is set to $D_o = 971$ μm. The droplet is initially stationary, and the surrounding gas is impulsively accelerated to a free-stream velocity of $U_\infty = 25$ m/s.

The dimensionless numbers relevant for thermal transport are the Reynolds number and the Prandtl number $Pr = c\mu/k$, where c is the specific heat capacity and k is the thermal conductivity. The Prandtl numbers of the droplet phase and gas phase are set to $Pr_L = 3.6$ and $Pr_g = 0.36$, respectively. The droplet is cool, with an initial temperature of $T_{L_o} = 300$ K, and the gas is hot, with an initial temperature of $T_{g_o} = 500$ K.

The initial droplet diameter D_o is resolved by 122 mesh spacings, and the droplet is placed in a tri-periodic domain with dimensions $6.62D_o \times 2.94D_o \times 2.94D_o$. The particle number density is $N_{pc} = 2$. The solution will be domain-dependent because the domain size is not significantly larger than the droplet. This is acceptable because the simulation serves solely as a demonstration.

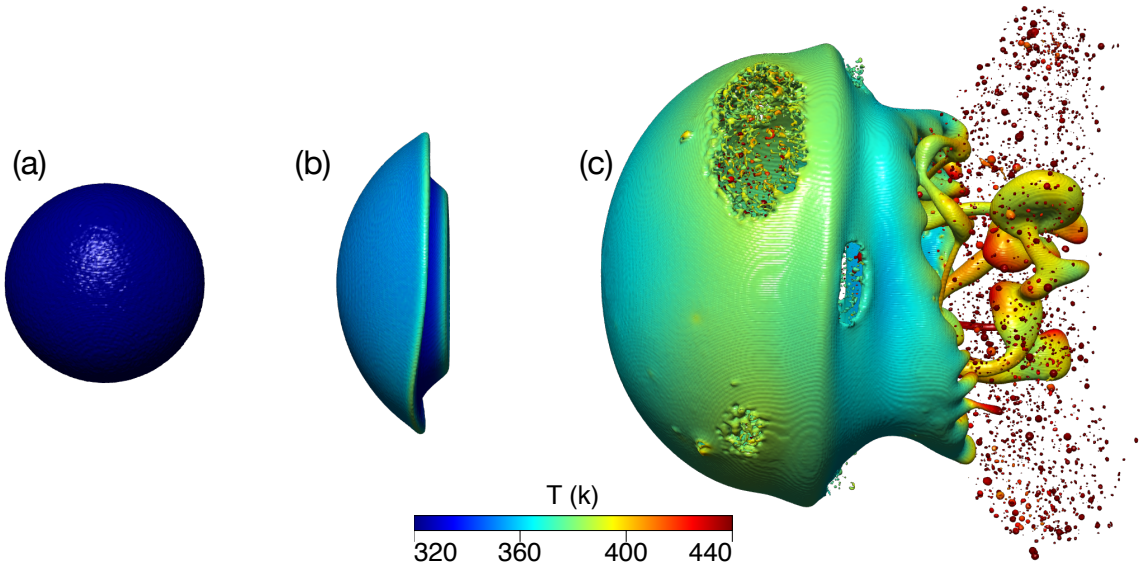

Figure 5.6. Temporal evolution of a droplet undergoing heat transfer and air blast atomization. Ambient air flows from left to right. (a) $t = 0$; (b) $t = 0.08$ ms; (c) $t = 0.3$ ms.

Results

Temporal evolution of the droplet deformation, breakup, and heating process is shown in Fig. 5.6 (the ambient air flows from left to right). The figure shows the droplet surface colored by the local temperature. Panel *a* shows the droplet in its initial condition, panel *b* shows the drop at $t = 0.08$ ms, and panel *c* shows the drop at $t = 0.3$ ms. The droplet in panel *a* is initially spherical and cool. As the fast moving gas passes over the droplet surface from left to right, a variety of dynamics evolve. Panel *b* shows that the droplet is simultaneously deformed and heated. At this early time, variations in temperature within the droplet phase are small, but slightly higher temperatures do appear at the outer-most surfaces on the downstream (right) side of the drop. At low density ratios, such as the present case with $\rho_L/\rho_g = 15$, a "bag" deformation is formed when the droplet wraps itself around a low pressure, recirculating wake. The early development of this bag deformation is apparent in

panel *b*, and the bag is fully developed at the later time in panel *c*. The apparent size of the droplet has increased significantly because it has been stretched into a thin film. As the droplet thins, holes form in the surface, resulting in the sheet breakup mechanism. Sheet breakup produces the small droplets that are prominent at the downstream side of the drop in panel *c*. Holes produced by sheet breakup grow radially under capillary forces, eventually merging with other nearby holes to form ligaments. The ligaments, also visible at the downstream side of the droplet, breakup under capillary forces, producing larger droplets. At the later time in panel *c*, significant temperature variations are apparent in the droplet phase. The smallest droplets produced by sheet breakup are very hot (> 440 k), while parts of the large structure remain relatively cool (< 360 k). This results from a variety of fluid-thermal interactions related to droplet and ligament surface areas and volumes, boundary layer thicknesses, and recirculation. We note that these temperatures determine evaporation rates in combustion systems, and therefore accurate thermal modeling in sprays is critical to predictive combustion simulation.

To emphasize that this arblast simulation was performed with the PMP method, we compare the droplet phase PMPs to their corresponding Eulerian representation on the mesh. Figure 5.7 panel *a* shows the droplet phase PMPs colored by temperature, and panel *b* shows the corresponding Eulerian data (reproduced from panel *c* in Fig. 5.6). The PMPs in panel *a* are the data used in the simulation to compute phase and thermal transport, while the Eulerian data in panel *b* is computed as a function of the PMPs on the Eulerian grid. In this simulation, each Eulerian control volume is discretized by two point mass particles. The particles are therefore a higher resolution discretization of the fluid system than the Eulerian grid. The capability of the PMPs to resolve more length scales than the Eulerian grid is made apparent by the larger number of small droplets in panel *a* than in panel *b*. In order to fully leverage the small scale data provided by the PMP method in multiphase simulations with large

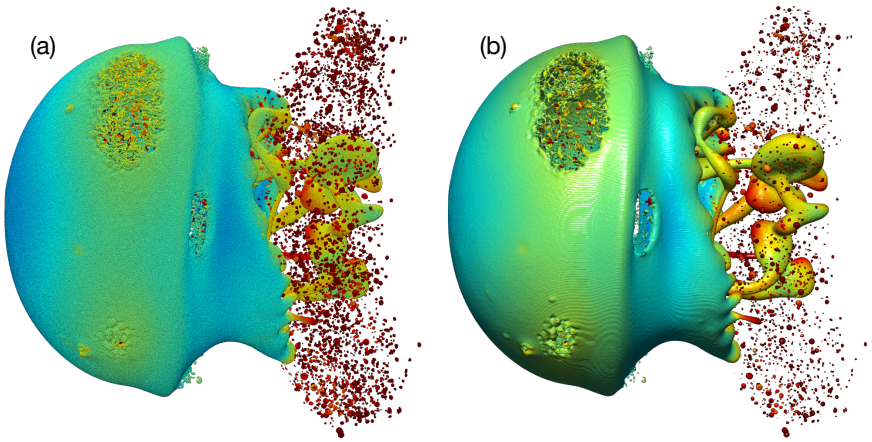

Figure 5.7. Comparison between droplet phase PMPs colored by temperature and the corresponding Eulerian data at time $t = 0.3$ ms. (a) PMP field; (b) Eulerian field.

property variations, we require a mass-momentum formulation that is consistent and conservative.

5.5 Chapter summary and conclusions

This chapter has extended the coupled Eulerian-Lagrangian PMP method to the simulation of scalar transport problems, presented specifically from the perspective of heat transfer. The motion of the PMPs provides consistent convection of energy and fluid properties, which is coupled to the diffusion operator of Cleary and Monaghan [16] via operator splitting. We have presented modifications to the diffusion operator that make it amenable to coupling with the Eulerian-Lagrangian PMP method.

Thermal transport has been verified in a number of heat transfer test problems. Thermal diffusion at the interface of hot water and cool air with an initially discontinuous temperature profile is demonstrated to produce long-time errors between 4% and 0.9% for the resolutions tested. The ability of the method to convect and diffuse an initially Gaussian temperature profile is assessed for high and low diffusivity fluids. Errors in the low diffusivity fluid grow in time independent of mesh resolution, but

remain below 2% for the cases tested. Increasing the mesh resolution decreases errors in the case of the high diffusivity fluid, with long-time errors dropping below 0.5% for all resolutions tested. The chapter concludes by demonstrating robustness via the simulation of heated air blast atomization of a droplet. In each of the considered test problems, the coupled Eulerian-Lagrangian PMP method for thermal transport conserves thermal energy and preserves boundedness of the temperature field. The robustness, accuracy, and simplicity of this approach makes it favorable for use in multiphase heat transfer problems.

Chapter 6

Consistent transport within the point mass particle framework

Inconsistencies in the transport of fluid properties and conserved quantities near fluid interfaces generate numerical instabilities [117, 106], as was discussed from the perspective of heat transfer in Chapter 5. Inconsistent transport arises from using different numerical schemes to compute the transport of different quantities. Inconsistent transport occurs, for example, when using a geometric VOF scheme to transport phase and density, and a centered scheme to transport momentum. In this chapter, we extend the PMP method to a conservative and consistent mass-momentum formulation by associating the particles with the fluid velocity. Formulations for consistent transport are outlined from the perspective of direct numerical simulation and large eddy simulation.

6.1 A fully consistent PMP

Consider a particle that moves in space according to

$$d\mathbf{X} = \mathbf{u}dt + \mathbf{U}dt + \frac{1}{2}\mathbf{A}dt^2 \tag{6.1}$$

where $\mathbf{X}$ is the location of the particle, $\mathbf{u}$ is the fluid velocity at the location of the particle, and $\mathbf{U}$ and $\mathbf{A}$ are the particle velocity and acceleration vectors (defined by the PMP method). The fluid velocity does not entirely determine the particle trajectory, and therefore the particle is *not* a fluid particle. The local fluid velocity $\mathbf{u}$ changes as the particle moves. In order to determine how the fluid velocity changes as the particle moves through space and time, we follow the derivation principles of Dreeben and Pope [24]. We consider a small increment of fluid velocity resulting from a small displacement of the particle in space and time:

$$d\mathbf{u} = \frac{\partial \mathbf{u}}{\partial t} dt + \nabla \mathbf{u} \cdot d\mathbf{X} + \left[\frac{\partial \mathbf{u}}{\partial t} dt \right] \left[\nabla \mathbf{u} \cdot d\mathbf{X} \right] + H.O.T. \tag{6.2}$$

Substituting Eq. (6.1) into Eq. (6.2), retaining terms up to $\mathcal{O}\left(dt^2\right)$ if they are multiplied by a component of acceleration $\mathbf{A}$, and up to $\mathcal{O}\left(dt\right)$ if they are multiplied by a component of either velocity $\mathbf{U}$ or $\mathbf{u}$, returns

$$d\mathbf{u} = \frac{\partial \mathbf{u}}{\partial t} dt + \nabla \mathbf{u} \cdot \left[\mathbf{u} dt + \mathbf{U} dt + \frac{1}{2} \mathbf{A} dt^2 \right]. \tag{6.3}$$

The rationale for retaining different time orders for accelerations and velocities follows from Eq. (6.1), where spatial displacement depends on velocity to first-order and on acceleration to second-order with respect to time. Partially expanding the right-hand-side of Eq. (6.3) returns

$$d\mathbf{u} = \frac{\partial \mathbf{u}}{\partial t} dt + \mathbf{u} \cdot \nabla \mathbf{u} dt + \nabla \mathbf{u} \cdot \left[\mathbf{U} dt + \frac{1}{2} \mathbf{A} dt^2 \right] \tag{6.4}$$

Again following [24], we consider the Navier-Stokes equations, expressed here in non-conservative form as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \left(\mu \left(\nabla \mathbf{u} + \nabla^T \mathbf{u} \right) \right). \tag{6.5}$$

Interfacial dynamics enter the Navier-Stokes equations through the jump condition on pressure at the phase interface,

$$[p] = \sigma \kappa + 2 \left[\mu \right] \hat{\mathbf{n}}^T \cdot \nabla \mathbf{u} \cdot \hat{\mathbf{n}}, \tag{6.6}$$

where $[p]$ is the pressure jump at the interface, σ is the surface tension coefficient, κ is the interfacial curvature, $[\mu] = \mu_L - \mu_g$ is the difference in fluid viscosities, and $\hat{\mathbf{n}}$ is the interfacial unit normal vector. The left-hand-side of Eq. (6.5) is proportional to the first two terms on the right-hand-side of Eq. (6.4). In Eq. (6.4) we substitute the pressure and viscous terms for the temporal and convective terms, resulting in

$$d\mathbf{u} = \left[-\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \left(\mu \left(\nabla \mathbf{u} + \nabla^T \mathbf{u} \right) \right) + \nabla \mathbf{u} \cdot \left[\mathbf{U} + \frac{1}{2}\mathbf{A}dt \right] \right] dt. \tag{6.7}$$

Equation (6.7) is a first-order evolution equation for the fluid velocity $\mathbf{u}$ evaluated at the location $\mathbf{X}$ of a particle that moves in space according to Eq. (6.1). In order to leverage Equation (6.7) in a coupled Eulerian-Lagrangian computational scheme, we require procedures to conservatively transfer information between the particles and an Eulerian grid.

6.1.1 Eulerian-Lagrangian communication

Each particle i has a mass M_i, a phase ζ_i^α, a position $\mathbf{X}_i$ that evolves according to Eq. (6.1), and a fluid velocity $\mathbf{u}_i$ that evolves according to Eq. (6.7). The momentum of particle i is then

$$\mathbf{G}_i = M_i \mathbf{u}_i, \tag{6.8}$$

where the momentum of the fluid system is defined by the sum of particle momenta.

Eulerian density and momentum fields are computed from the particles. Each particle distributes its mass and momentum to an Eulerian grid using a spatially-compact distribution function $\mathcal{G}$. The function $\mathcal{G}$ is compact in that it only distributes mass and momentum to Eulerian cells within a cut-off distance h_d of particle i. In order to ensure consistency on a staggered Eulerian flow solver grid, mass and momentum must be defined at the same spatial locations [117, 106]. Similar to prior work [117, 106], we utilize a sub-grid with twice the resolution of the flow solver grid – in

three dimensions, every Eulerian control volume contains eight sub-grid nodes. Unlike prior work, the sub-grid is a collocated grid, rather than a staggered grid. Each of the collocated sub-grid nodes gathers mass and momentum information from the particles, and then transfers the mass and momentum to the parent staggered-grid control volume. The details of these communications are described in this section.

The mass distributed from particle i to sub-grid node k, denoted by $M_{i\to k}$, is computed according to

$$M_{i\to k} = \frac{M_i \mathcal{G}\left(\mathbf{X}_i - \mathbf{x}_k, h_d\right)}{\sum_{j=1}^{N_j} \mathcal{G}\left(\mathbf{X}_i - \mathbf{x}_j, h_d\right)}. \tag{6.9}$$

The denominator on the right-hand-side is a scaling factor that ensures the totality of M_i is distributed amongst the N_j sub-grid nodes within the non-zero distribution radius h_d of particle i. The total mass distributed from particles to sub-grid node k, $\overline{m}_k$, is computed by summing the mass contributions from all particles,

$$\overline{m}_k = \sum_{i=1}^{N_i} M_{i\to k} = \sum_{i=1}^{N_i} \frac{M_i \mathcal{G}\left(\mathbf{X}_i - \mathbf{x}_k, h_d\right)}{\sum_{j=1}^{N_j} \mathcal{G}\left(\mathbf{X}_i - \mathbf{x}_j, h_d\right)}, \tag{6.10}$$

where N_i is the number of particles for which $M_{i\to k} \neq 0$. An identical distribution is performed for the momentum

$$\left(\overline{m\mathbf{u}}\right)_k = \sum_{i=1}^{N_i} \left(M\mathbf{u}\right)_{i\to k} = \sum_{i=1}^{N_i} \frac{M_i \mathbf{u}_i \mathcal{G}\left(\mathbf{X}_i - \mathbf{x}_k, h_d\right)}{\sum_{j=1}^{N_j} \mathcal{G}\left(\mathbf{X}_i - \mathbf{x}_j, h_d\right)}, \tag{6.11}$$

where $\left(\overline{m\mathbf{u}}\right)_k$ is the momentum distributed from the particles to sub-grid node k. Mass and momentum are next transferred from the sub-grid to the flow solver grid.

Each sub-grid node falls within the bounds of a momentum control volume. The density and momentum are computed at the centroid of each momentum control volume by volume-averaging the sub-grid nodal values. This procedure returns the density according to

$$\rho = \frac{1}{V} \sum_{k=1}^{N_k} \overline{m}_k = \frac{1}{V} \sum_{k=1}^{N_k} \sum_{i=1}^{N_i} \frac{M_i \mathcal{G}\left(\mathbf{X}_i - \mathbf{x}_k, h_d\right)}{\sum_{j=1}^{N_j} \mathcal{G}\left(\mathbf{X}_i - \mathbf{x}_j, h_d\right)}, \tag{6.12}$$

where $V = \Delta x^3$ is the volume and $N_k = 8$ is the number of sub-grid nodes within the control volume (in three-dimensions). The momentum on the flow solver grid is similarly computed,

$$\mathbf{g} = \frac{1}{V}\sum_{k=1}^{N_k}\left(\overline{m\mathbf{u}}\right)_k = \frac{1}{V}\sum_{k=1}^{N_k}\sum_{i=1}^{N_i}\frac{M_i\mathbf{u}_i\mathcal{G}\left(\mathbf{X}_i - \mathbf{x}_k, h_d\right)}{\sum_{j=1}^{N_j}\mathcal{G}\left(\mathbf{X}_i - \mathbf{x}_j, h_d\right)}. \tag{6.13}$$

Equations (6.12) and (6.13) provide an Eulerian density and momentum as a function of the particle field.

We now consider communication in the opposite direction – from the grid to the particles. Consider an increment of momentum $\delta\mathbf{g}$ computed on the Eulerian grid. We require a conservative mechanism to increment the momenta of the particles. For this purpose we again consider a distribution function. The momentum increment on the flow solver grid is first conservatively transferred to the sub-grid nodes. A simple approach is to divide the momentum increment uniformly over the sub-grid nodes N_k that were originally used to generate $\mathbf{g}$ in Eq. (6.13)

$$\delta\left(\overline{m\mathbf{u}}\right)_k = \delta\mathbf{g}/N_k. \tag{6.14}$$

The sub-grid nodal momenta can then be distributed to the particles in proportion to the mass the particles contributed to each sub-grid node

$$\delta\mathbf{u}_i = \sum_{k=1}^{N_k}\frac{\delta\left(\overline{m\mathbf{u}}\right)_k M_{i\to k}}{M_i\overline{m}_k}. \tag{6.15}$$

The communication procedures described in this section allow for conservative and consistent transfer of mass and momentum information between the particle field and a staggered Eulerian grid. These procedures are the building blocks of a coupled Eulerian-Lagrangian flow solving scheme that is conservative and consistent.

6.1.2 Discrete mass and momentum integration

The fluid velocity $\mathbf{u}_i$, and the particle position $\mathbf{X}_i$, velocity $\mathbf{U}_i$, and acceleration $\mathbf{A}_i$ are known at time level n. The objective is to advance the solution to time level

$n+1$, a finite increment in time of Δt, while conserving mass and momentum for arbitrary property variations amongst the phases. The solution is advanced with a combination of Lagrangian and Eulerian operations described in this section.

The particles move from $\mathbf{X}_i^n$ to $\mathbf{X}_i^{n+1}$ with the known fluid velocity $\mathbf{u}_i^n$ according to Eq. (6.1). After the particles have been moved to their new location, the Eulerian density at time level $n+1$ is computed according to Eq. (6.12)

$$\rho^{n+1} = \frac{1}{V}\sum_{k=1}^{N_k} \overline{m}_k^{n+1} = \frac{1}{V}\sum_{k=1}^{N_k}\sum_{i=1}^{N_i} \frac{M_i \mathcal{G}\left(\mathbf{X}_i^{n+1} - \mathbf{x}_k, h_d\right)}{\sum_{j=1}^{N_j} \mathcal{G}\left(\mathbf{X}_i^{n+1} - \mathbf{x}_j, h_d\right)}. \tag{6.16}$$

Moving the particles naturally accounts for convection of momentum through the Lagrangian operation $\mathcal{C}$, represented here by

$$\mathcal{C} = \mathbf{g}^n - \nabla \cdot \left(\rho \mathbf{u} \otimes \mathbf{u}\right)^n \Delta t. \tag{6.17}$$

As the particles move from $\mathbf{X}_i^n$ to $\mathbf{X}_i^{n+1}$, the fluid velocity they carry is modified due to displacement from the particle velocity $\mathbf{U}_i^n$ and acceleration $\mathbf{A}_i^n$. This modification is accounted for by the last two terms on the right-hand-side of Eq. (6.7). These terms are non-conservative with respect to momentum. They are made conservative by considering an ensemble of N_e particles, comprised of the particles contained in a control volume that belong to the same phase. The mean $\mathbf{E}_{nc}$ of the non-conservative terms is computed amongst the ensemble

$$\mathbf{E}_{nc} = \frac{1}{N_e}\sum_{j=1}^{N_e} \left(\mathbf{U}_j^n + \frac{1}{2}\mathbf{A}_j^n \Delta t\right) \cdot \nabla \mathbf{u}^n \left(\mathbf{X}_j^n\right) \Delta t, \tag{6.18}$$

where $\nabla \mathbf{u}^n \left(\mathbf{X}_j^n\right)$ is the gradient of Eulerian velocity at time level n interpolated to the particle location at time level n. Subtracting $\mathbf{E}_{nc}$ from the non-conservative terms negates the conservation error. The resultant expression for fluid velocity on particle i at position $\mathbf{X}_i^{n+1}$ is

$$\mathbf{u}_i^* = \mathbf{u}_i^n + \left(\mathbf{U}_i^n + \frac{1}{2}\mathbf{A}_i^n \Delta t\right) \cdot \nabla \mathbf{u}^n \left(\mathbf{X}_i^n\right) \Delta t - \mathbf{E}_{nc}, \tag{6.19}$$

where $\mathbf{u}_i^*$ is the fluid velocity at time level n, augmented by the displacement in space due to $\mathbf{U}_i^n$ and $\mathbf{A}_i^n$. The Lagrangian convection operator $\mathcal{C}$ is then computed according to

$$\begin{aligned}\mathcal{C} =&\mathbf{g}^n - \nabla\cdot\left(\rho\mathbf{u}\otimes\mathbf{u}\right)^n\Delta t\\ =&\frac{1}{V}\sum_{k=1}^{N_k}\left(\overline{m\mathbf{u}}\right)_k^* = \frac{1}{V}\sum_{k=1}^{N_k}\sum_{i=1}^{N_i}\frac{M_i\mathbf{u}_i^*\mathcal{G}\left(\mathbf{X}_i^{n+1}-\mathbf{x}_k,h_d\right)}{\sum_{j=1}^{N_j}\mathcal{G}\left(\mathbf{X}_i^{n+1}-\mathbf{x}_j,h_d\right)}.\end{aligned} \tag{6.20}$$

Moving the particles to $\mathbf{X}_i^{n+1}$ and updating $\mathbf{u}_i^n$ to $\mathbf{u}_i^*$ only partially advances the solution to time level $n+1$: viscous, pressure, and surface tension effects are treated with the Eulerian grid.

Consider an explicit, first-order discretization of the momentum equation with respect to time

$$\frac{\mathbf{g}^{n+1}-\mathbf{g}^n}{\Delta t} = -\nabla\cdot\left(\rho\mathbf{u}\otimes\mathbf{u}\right)^n + \nabla\cdot\left(2\mu\mathbf{S}-p\mathbf{I}\right)^n, \tag{6.21}$$

where the effects of surface tension have been absorbed by the pressure p according to Eq. (6.6). Solving for the momentum at time level $n+1$ returns

$$\mathbf{g}^{n+1} = \mathbf{g}^n - \nabla\cdot\left(\rho\mathbf{u}\otimes\mathbf{u}\right)^n\Delta t + \nabla\cdot\left(2\mu\mathbf{S}-p\mathbf{I}\right)^n\Delta t. \tag{6.22}$$

The effects of convection have already been computed via the Lagrangian operator $\mathcal{C}$, allowing us to write

$$\mathbf{g}^{n+1} = \mathcal{C} + \nabla\cdot\left(2\mu\mathbf{S}-p\mathbf{I}\right)^n\Delta t. \tag{6.23}$$

We also choose to re-cast the remaining terms as a function of an intermediate velocity $\mathbf{u}^{**}$, defined by the most up-to-date particle information

$$\mathbf{u}^{**} = \frac{\mathcal{C}}{\rho^{n+1}}, \tag{6.24}$$

resulting in

$$\mathbf{g}^{n+1} = \mathcal{C} + \nabla\cdot\left(2\mu\mathbf{S}-p\mathbf{I}\right)^{**}\Delta t. \tag{6.25}$$

This equation is solved for an incompressible flow with standard discretizations for the viscous term, operator splitting for pressure, and a ghost-fluid method for surface tension.

After the Eulerian momentum is updated to $\mathbf{g}^{n+1}$, a final modification of the fluid velocity carried by the particles is required. Referencing Eq. (6.7), we note that the last two terms on the right-had side have already been accounted for during the particle displacement. Velocity changes due to pressure, surface tension, and viscous forces must be considered next. Using Eq. (6.7) and Eq. (6.14), the relevant momentum increment for the sub-grid nodes is

$$\delta\left(\overline{m}\mathbf{u}\right)_k^{n+1} = \nabla\cdot\left(2\mu\mathbf{S} - p\mathbf{I}\right)^{**}\Delta t/N_k. \tag{6.26}$$

The increment to particle velocity, according to Eq. (6.15), is

$$\delta\mathbf{u}_i^{n+1} = \sum_{k=1}^{N_k}\frac{\delta\left(\overline{m}\mathbf{u}\right)_k^{n+1}M_{i\to k}^{n+1}}{M_i\overline{m}_k^{n+1}}, \tag{6.27}$$

and the updated particle velocity is given by

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \delta\mathbf{u}_i^{n+1}. \tag{6.28}$$

The procedures outlined in this section provide a conservative and consistent method for integrating the multiphase Navier-Stokes equations with an arbitrary number of phases and arbitrary property variations. Simple multiphase flows can be fully-resolved on the Eulerian mesh. Complicated turbulent flows, however, contain a wide range of spatio-temporal scales. These flows require the use of LES.

6.2 Multiphase LES with the PMP

Large eddy simulation reduces the computational cost of predicting fluid motion by solving filtered transport equations. The filtered fields have – ideally – a statistical

relation to a filtered realization of the turbulent flow [110]. The Reynolds filtering operator, denoted by $\langle\rangle_\ell$, is defined by

$$\langle f(\mathbf{x},t)\rangle_\ell = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\mathbf{x}',t')\,G(\mathbf{x}-\mathbf{x}',t-t')\,dt'd\mathbf{x}', \tag{6.29}$$

where f is a function of space and time and G is the filter. The filter must satisfy a number of properties, and information pertinent to multiphase filtering operators can be found throughout the literature [55, 72, 14]. The Favre filtering operator is defined by

$$\langle f\rangle_L = \frac{\langle \rho f\rangle_\ell}{\langle\rho\rangle_\ell}, \tag{6.30}$$

where ρ is the fluid density. Favre filtering therefore produces a function that is both filtered and density-weighted. The purpose of the Favre filter is to decompose the filtered momentum into a Favre filtered velocity and a filtered density according to the relation

$$\langle\rho\mathbf{u}\rangle_\ell = \langle\rho\rangle_\ell\frac{\langle\rho\mathbf{u}\rangle_\ell}{\langle\rho\rangle_\ell} = \langle\rho\rangle_\ell\langle\mathbf{u}\rangle_L, \tag{6.31}$$

where $\mathbf{u}$ is the velocity vector.

The Favre filtered governing equations for multiphase fluid dynamics are derived by first operating a Reynolds filter on the multiphase Navier-Stokes equations, followed by manipulating the equations to express all velocities as Favre filtered velocities [72]. The resultant expression for mass conservation is

$$\frac{\partial\langle\rho\rangle_\ell}{\partial t} + \nabla\cdot(\langle\rho\rangle_\ell\langle\mathbf{u}\rangle_L) = 0, \tag{6.32}$$

where (in the case of an incompressible fluid) the divergence of the Favre filtered velocity field is constrained by

$$\nabla\cdot\langle\mathbf{u}\rangle_L = \tau_{un} \tag{6.33}$$

where

$$\tau_{un} = \frac{\langle\mathbf{u}\cdot\nabla\rho\rangle_\ell - \langle\mathbf{u}\rangle_L\cdot\nabla\langle\rho\rangle_\ell}{\langle\rho\rangle_\ell}. \tag{6.34}$$

The filtered momentum equation is given by

$$\begin{aligned}\frac{\partial\langle\rho\rangle_\ell\langle\mathbf{u}\rangle_L}{\partial t}+\nabla\cdot(\langle\rho\rangle_\ell\langle\mathbf{u}\rangle_L\otimes\langle\mathbf{u}\rangle_L)=\\ \nabla\cdot(2\langle\mu\rangle_\ell\langle\mathbf{S}\rangle_L-\langle p\rangle_\ell\mathbf{I})-\nabla\cdot(\tau_{\rho uu}-\tau_{\mu\mathbf{S}})+\langle\mathbf{F}^s\rangle_\ell,\end{aligned} \tag{6.35}$$

where μ is the dynamic viscosity, $\mathbf{S}$ is the strain rate tensor, p is the pressure, and $\mathbf{I}$ is the identity matrix. The sub-filter and multiphase terms are the residual convective stress tensor

$$\tau_{\rho uu}=\langle\rho\rangle_\ell\left[\langle\mathbf{u}\otimes\mathbf{u}\rangle_L-\langle\mathbf{u}\rangle_L\otimes\langle\mathbf{u}\rangle_L\right], \tag{6.36}$$

the residual viscous stress tensor

$$\tau_{\mu\mathbf{S}}=2\left[\langle\mu\mathbf{S}\rangle_\ell-\langle\mu\rangle_\ell\langle\mathbf{S}\rangle_L\right], \tag{6.37}$$

and the filtered surface tension

$$\langle\mathbf{F}^s\rangle_\ell=\langle\sigma\kappa\hat{\mathbf{n}}\delta\rangle_\ell, \tag{6.38}$$

where sub-filter notation is taken from Larocque et al. [72]. The filtered surface tension acts at the interface δ in the normal direction $\hat{\mathbf{n}}$, and is proportional to the surface tension coefficient σ and the interfacial curvature κ. The final multiphase transport equation is for the phase indicator function of phase ζ^α

$$\frac{\partial\langle\chi^\alpha\rangle_\ell}{\partial t}+\nabla\cdot(\langle\chi^\alpha\rangle_\ell\langle\mathbf{u}\rangle_L)=-\nabla\cdot\tau_{\chi^\alpha}, \tag{6.39}$$

where χ^α is the phase indicator, and

$$\tau_{\chi^\alpha}=\langle\chi^\alpha\mathbf{u}\rangle_\ell-\langle\chi^\alpha\rangle_\ell\langle\mathbf{u}\rangle_L \tag{6.40}$$

is the residual flux of the phase indicator.

The multiphase LES equations, as expressed here, include five sub-grid-scale terms: τ_{un}, $\tau_{\rho uu}$, $\tau_{\mu\mathbf{S}}$, $\langle\mathbf{F}^s\rangle_\ell$, and τ_{χ^α}. Unlike single phase fluid dynamics, multiphase interfacial dynamics are not cascades; the small scale features are not necessarily

functions of resolved features. This topic is considered in depth by Herrmann and Gorokhovski [54], who argue that a fully-resolved interface is required for predictive LES of multiphase systems because breakup processes in the sub-grid cannot be informed by the resolved scales. In other words, Herrmann and Gorokhovski [54] argue it is insufficient to simply close the SGS terms. Rather, it is as important to maintain a physical representation of the sub-grid scales because the smallest scales in multiphase flows are typically of interest in applications. In the following sections, we extend the PMP approach to LES by leveraging concepts used in probability density function and filtered density function methods.

6.2.1 PMP turbulence modeling with the Langevin equation

Decompose the fluid particle velocity equation, given by Eq. (6.7), into filtered and residual components

$$\begin{aligned} d\mathbf{u} = &-\frac{1}{\langle\rho\rangle_\ell}\nabla\langle p\rangle_\ell dt + \frac{1}{\langle\rho\rangle_\ell}\nabla\cdot\left(\langle\mu\rangle_\ell\left(\nabla\langle\mathbf{u}\rangle_L + \nabla^T\langle\mathbf{u}\rangle_L\right)\right)dt + \\ &\nabla\langle\mathbf{u}\rangle_L\cdot\left[\mathbf{U} + \frac{1}{2}\mathbf{A}dt\right]dt + \\ &\tau_1 dt + \tau_2 dt + \tau_3 dt, \end{aligned} \tag{6.41}$$

where τ_1, τ_2, and τ_3 are residual terms defined by

$$\tau_1 = -\frac{1}{\rho}\nabla p + \frac{1}{\langle\rho\rangle_\ell}\nabla\langle p\rangle_\ell, \tag{6.42}$$

and

$$\tau_2 = \frac{1}{\rho}\nabla\cdot\left(\mu\left(\nabla\mathbf{u} + \nabla^T\mathbf{u}\right)\right) - \frac{1}{\langle\rho\rangle_\ell}\nabla\cdot\left(\langle\mu\rangle_\ell\left(\nabla\langle\mathbf{u}\rangle_L + \nabla^T\langle\mathbf{u}\rangle_L\right)\right), \tag{6.43}$$

and

$$\tau_3 = \nabla\mathbf{u}\cdot\left[\mathbf{U} + \frac{1}{2}\mathbf{A}dt\right] - \nabla\langle\mathbf{u}\rangle_L\cdot\left[\mathbf{U} + \frac{1}{2}\mathbf{A}dt\right]. \tag{6.44}$$

The principal challenge of performing LES with the PMP method is closing each of these residual terms – this remains an open problem with a variety of potential solutions. One solution is to use fractal interpolation [121] to produce a synthetic velocity field at the resolution of the particle field. This approach has been proposed in the context of Eulerian VOF simulations, and requires the phase interface to be fully-resolved [65]. A second potential solution is to model the residual terms using the accelerations acquired from the PMP method, or stochastic models for fluid particle acceleration (such as those discussed in [111]). In this section, in order to adhere to prior work in probability density function methods [46, 24], we close the unknown residual quantities with the Generalized Langevin Equation

$$(\tau_1 + \tau_2 + \tau_3)\, dt = \mathbf{G}\left(\mathbf{u} - \langle \mathbf{u} \rangle_L\right) dt + \sqrt{C_0 \epsilon} d\mathbf{W} \tag{6.45}$$

where $\mathbf{G}$, C_0, and ϵ are parameters of the Generalized Langevin Equation (to be defined shortly), and $d\mathbf{W}$ is an increment of the Wiener vector process. The governing equations for the modeled particles are then

$$d\mathbf{X} = \mathbf{u}dt + \mathbf{U}dt + \frac{1}{2}\mathbf{A}dt^2 \tag{6.46}$$

and

$$\begin{aligned} d\mathbf{u} = &-\frac{1}{\langle \rho \rangle_\ell} \nabla \langle p \rangle_\ell dt + \frac{1}{\langle \rho \rangle_\ell} \nabla \cdot \left(\langle \mu \rangle_\ell \left(\nabla \langle \mathbf{u} \rangle_L + \nabla^T \langle \mathbf{u} \rangle_L \right) \right) dt + \\ &\nabla \langle \mathbf{u} \rangle_L \cdot \left[\mathbf{U} + \frac{1}{2}\mathbf{A}dt \right] dt + \\ &\mathbf{G}\left(\mathbf{u} - \langle \mathbf{u} \rangle_L\right) dt + \sqrt{C_0 \epsilon} d\mathbf{W}, \end{aligned} \tag{6.47}$$

which determine the evolution of particle position and fluid velocity in time. The system of equations can be closed by specifying the simplified Langevin model, where

$$\mathbf{G} = -\omega\left(\frac{1}{2}+\frac{3}{4}C_0\right)\mathbf{I}, \tag{6.48}$$

$$\omega = \frac{\epsilon}{k}, \tag{6.49}$$

$$\epsilon = C_\epsilon \frac{k^{3/2}}{\Delta x}, \tag{6.50}$$

$$k = \frac{1}{2}\left[\langle \mathbf{u}\cdot\mathbf{u}\rangle_L - \langle \mathbf{u}\rangle_L\langle \mathbf{u}\rangle_L\right], \tag{6.51}$$

where C_0 and C_ϵ are model parameters.

6.2.2 Discrete mass and momentum integration

The solution procedure for the filtered equations is similar to that outlined in Section 6.1.2 for the non-filtered equations. This section outlines some notable exceptions and caveats.

The particles, which are significantly more numerous than the number of Eulerian grid points, are advanced from position $\mathbf{X}_i^n$ to $\mathbf{X}_i^{n+1}$ with velocity $\mathbf{u}_i^n$ (and $\mathbf{U}_i^n$ and $\mathbf{A}_i^n$), and the filtered density is computed at $n+1$

$$\langle\rho\rangle_\ell^{n+1} = \frac{1}{V}\sum_{k=1}^{N_k}\langle\overline{m}_k\rangle_\ell^{n+1} = \frac{1}{V}\sum_{k=1}^{N_k}\sum_{i=1}^{N_i}\frac{M_i\mathcal{G}\left(\mathbf{X}_i^{n+1}-\mathbf{x}_k,h_d\right)}{\sum_{j=1}^{N_j}\mathcal{G}\left(\mathbf{X}_i^{n+1}-\mathbf{x}_j,h_d\right)}. \tag{6.52}$$

The intermediate fluid velocities $\mathbf{u}_i^*$ are computed next. In the context of LES, the intermediate fluid velocity includes the spatial corrections due to displacement via particle velocity, in addition to residual terms introduced by the Langevin equation. The last term on the right-hand-side of Eq. (6.47) is a non-conservative, random walk in velocity space. This term is included in the non-conservative correction

$$\mathbf{E}_{nc} = \frac{1}{N_e}\sum_{j=1}^{N_e}\left[\left(\mathbf{U}_j^n+\frac{1}{2}\mathbf{A}_j^n\Delta t\right)\cdot\nabla\langle\mathbf{u}\rangle_L^n\left(\mathbf{X}_j^n\right)\Delta t+\sqrt{C_0\epsilon}d\mathbf{W}_j\right], \tag{6.53}$$

where $\nabla\langle\mathbf{u}\rangle_L^n\left(\mathbf{X}_j^n\right)$ is the gradient of the Favre-filtered Eulerian velocity at time level n interpolated to the particle location at time level n. The resultant expression for fluid velocity on particle i at position $\mathbf{X}_i^{n+1}$ is

$$\begin{aligned}\mathbf{u}_i^* =&\mathbf{u}_i^n + \left(\mathbf{U}_i^n + \frac{1}{2}\mathbf{A}_i^n\Delta t\right)\cdot\nabla\langle\mathbf{u}\rangle_L^n\left(\mathbf{X}_i^n\right)\Delta t+\\ &\mathbf{G}\left(\mathbf{u}_i^n - \langle\mathbf{u}\rangle_L^n\right)\Delta t + \sqrt{C_0\epsilon}d\mathbf{W}_i - \mathbf{E}_{nc}.\end{aligned} \tag{6.54}$$

The filtered Lagrangian convection operator $\langle\mathcal{C}\rangle_\ell$ is then computed according to

$$\langle\mathcal{C}\rangle_\ell = \frac{1}{V}\sum_{k=1}^{N_k}\langle\overline{m\mathbf{u}}\rangle_{\ell k}^* = \frac{1}{V}\sum_{k=1}^{N_k}\sum_{i=1}^{N_i}\frac{M_i\mathbf{u}_i^*\mathcal{G}\left(\mathbf{X}_i^{n+1}-\mathbf{x}_k,h_d\right)}{\sum_{j=1}^{N_j}\mathcal{G}\left(\mathbf{X}_i^{n+1}-\mathbf{x}_j,h_d\right)}. \tag{6.55}$$

The filtered momentum equation is again considered in discrete, first-order fashion with respect to time

$$\frac{\langle\rho\mathbf{u}\rangle_\ell^{n+1}-\langle\rho\mathbf{u}\rangle_\ell^n}{\Delta t} = -\nabla\cdot\langle\rho\mathbf{u}\otimes\mathbf{u}\rangle_\ell^n + \nabla\cdot\left(2\langle\mu\rangle_\ell\langle\mathbf{S}\rangle_L - \langle p\rangle_\ell\mathbf{I}\right)^n + \nabla\cdot\tau_{\mu\mathbf{S}}^n + \langle\mathbf{F}^s\rangle_\ell^n. \tag{6.56}$$

The residual inertial term $\tau_{\rho uu}$ that typically requires modeling does not appear here because the filtered momentum has not been decomposed into velocity and density components. Instead, the effects of this term have been absorbed into the filtered Lagrangian convection operator by means of the Langevin equation. Solving for the filtered momentum at $n+1$ and substituting the filtered Lagrangian convection operator returns

$$\langle\rho\mathbf{u}\rangle_\ell^{n+1} = \langle\mathcal{C}\rangle_\ell + \nabla\cdot\left(2\langle\mu\rangle_\ell\langle\mathbf{S}\rangle_L - p\mathbf{I}\right)^{**} + \nabla\cdot\tau_{\mu\mathbf{S}}^n + \langle\mathbf{F}^s\rangle_\ell^n, \tag{6.57}$$

where the Favre-filtered velocity used in the strain rate tensor is computed by

$$\langle\mathbf{u}\rangle_L^{**} = \frac{\langle\mathcal{C}\rangle_\ell}{\langle\rho\rangle_\ell^{n+1}}. \tag{6.58}$$

The residual inertial term $\tau_{\rho uu}$ has been closed, but we have arrived at a series of challenges:

1. Closure of the SGS viscous term $\tau^n_{\mu\mathbf{S}}$ requires modeling decisions. This term is much less important than all of the other SGS terms, and it is only non-zero at the interface [72]. For now, without a rational closure strategy, this term is simply neglected.

2. Closure of the filtered surface tension $\langle \mathbf{F}^s \rangle_\ell$ is required for solution of the Poisson equation for pressure. A number of potential closures are available. Following the work of [54], one could compute the surface tension on every PMP near the interface via the Finite Particle Method [148], followed by an explicit filtering operation, resulting in $\langle \mathbf{F}^s \rangle_\ell$. This is a mathematically rigorous closure (if the particle field is sufficiently resolved, as is required by [54]). The open question is how well the approach works in practice. Alternatively, the surface tension computed from the pairwise force method could be used for the same purpose. Evaluating the relative merits of these closures as a function of mesh resolution and filter size is an area of future work.

3. The Favre-filtered velocity field is not divergence-free at the interface, which precludes the use of the standard Poisson approach for identifying the pressure.

An immediate solution is simply to neglect $\tau^n_{\mu\mathbf{S}}$, compute $\langle \mathbf{F}^s \rangle_\ell$ via explicit filtering, and to assume the Favre-filtered velocity field is divergence-free everywhere. Equation (6.57) is then solved with the standard operator splitting approach for pressure, coupled with the ghost fluid method for surface tension. Momentum increments are then transferred to the particles with expressions analogous to Eqns. (6.26)-(6.28). This LES framework enforces conservative and consistent transport of all conserved quantities.

6.3 Chapter summary and conclusions

This chapter has outlined a new framework for simulating interfacial flows. In order to address the consistency and stability challenges of spray simulation, we have extended the coupled Eulerian-Lagrangian PMP formulation to a completely consistent and conservative mass-momentum implementation. This extension required the derivation of a governing equation for fluid velocity in the context of the PMP, as well as a discrete integration strategy for the Naiver-Stokes equations. Additionally, in order to address the challenge of resolving the spatio-temporal scales in a spray, we have extended the consistent mass-momentum framework to a large eddy simulation implementation. The LES implementation is favorable because it preserves all of the conservation and consistency properties of the PMP. With appropriate selection of closure models, the LES implementation has the potential to become a robust tool for performing predictive LES of sprays.

Chapter 7

Conclusion

7.1 Summary

This dissertation has presented methods for the modeling and simulation of interfacial flows, with specific emphasis on developing methods for improving the predictive capability of spray simulation. Topics covered include methods for computing curvature, tracking the location of phase interfaces, and computing multiphase heat transfer. Frameworks are also proposed for consistently and conservatively solving the multiphase Navier-Stokes equations and the multiphase large eddy simulation equations. This chapter presents a general summary of the work presented in this dissertation, while detailed summaries of work performed appear at the end of each chapter.

We begin by deriving and evaluating two finite particle method (FPM) approaches for computing curvature of poorly resolved interfaces in the context of volume of fluid simulations. Performance of the FPM methods are evaluated via the L_2 and L_∞ error norms corresponding to the computed curvature of spheres. All FPM schemes are shown to out-perform the mesh decoupled height function method at low resolutions. At high resolutions, however, the height function method converges, whereas the FPM

approaches do not converge for a fixed influence radius – convergence can be achieved by modifying the influence radius as a function of resolution. Dynamic performance of the FPM is evaluated by simulating the shape oscillations of initially elliptical ethanol droplets suspended in air. The oscillation frequencies of the simulated droplets are compared to the analytical oscillation frequency. Results show that the length scales used to compute the interfacial curvature should be comparable to the length scales of the interface. Two adaptive length scale FPM approaches are presented, and both approaches out-perform the standard height function method in the oscillating ethanol droplet test problem at resolutions relevant to spray simulation; the adaptive FPM approach is therefore a valuable new tool for use in the simulation of sprays.

After qualifying the FPM, we present a coupled Eulerian-Lagrangian approach for the modeling and simulation of immiscible multiphase flow systems. The Naiver-Stokes equations are solved on a traditional Eulerian grid while the fluid mass and phase information is discretized by Lagrangian point mass particles. The method is novel because the particles move with a velocity that enforces consistency between the particle field density and the fluid density. The approach is advantageous in that (i) an arbitrary number of phases are easily represented, (ii) the particles remain well-distributed in space, even near merging and diverging characteristics, (iii) mass conservation is easily controlled, and (iv) the methodology is applicable to a wide range of Courant numbers. The governing equations are derived and a numerical method is presented that is applicable to incompressible flows. Performance is assessed via standard two-dimensional and three-dimensional phase transport tests as a function of both Eulerian grid resolution and Lagrangian particle resolution. Results show that the shape error converges with first-order with respect to increasing either Eulerian grid resolution or particle resolution, while mass conservation errors converge with the square root. The method is particularly applicable to sprays because it out-performs

contemporary volume of fluid methods with respect to shape errors at the low resolutions relevant to spray simulation. The method is shown to successfully simulate expanding elliptical regions, stationary and oscillating droplets, a droplet in shear flow, and a Rayleigh-Taylor instability. The Eulerian-Lagrangian PMP method is then extended to a framework for consistently solving multiphase convection-diffusion problems with discontinuous property variations, presented specifically from the perspective of heat transfer. The convection-diffusion scheme is demonstrated to successfully quantify diffusion rates at an air-water interface, and to quantify the heat transfer dynamic in a convecting and diffusing Gaussian temperature distribution. Robustness is demonstrated via the heated air blast atomization of a droplet. In each heat transfer case, the method conserves thermal energy and preserves boundedness of the temperature field.

We finally proceed to the proposal of a mass-momentum consistent extension of the coupled Eulerian-Lagrangian PMP method for the solution of the Navier-Stokes equations. Consistent transport is required for stable, robust simulation of multiphase flow problems with discontinuous properties. The governing equations are derived, and a discrete time-integration scheme is introduced. We conclude by proposing a mass-momentum consistent scheme for performing LES of sprays, with a discussion of required closures.

7.2 Ongoing and future work

A variety of efforts are on-going, or proposed as future work:

- Current efforts are underway by co-workers to further quantify performance of the coupled Eulerian-Lagrangian PMP scheme for solving convection-diffusion problems. Specifically, heat transfer to gas bubbles rising in oil and thermal boundary layer development over flat plates are being examined. Thank you to

Mustafa Kaddoura of the University of Minnesota for carrying on these efforts.

- Performance of the mass-momentum consistent Eulerian-Lagrangian PMP scheme is currently being quantified as a function of numerical parameters in three test problems: the convection of an inviscid liquid suspended in inviscid gas; the shape oscillation of droplets due to surface tension, considered with and without viscosity; and the evolution of a single phase inviscid flow with an analytical solution. Results are very promising, but considerable effort must be spent to quantify the dependence of performance on the myriad of decisions required for numerical implementation.

- After quantifying performance of the mass-momentum consistent Eulerian-Lagrangian PMP, attention should be given to assessing the viability of the LES framework for spray simulation outlined at the end of Chapter 6. Many of the challenges of this implementation are outlined in Section 6.2.2, and each of these challenges should be considered in detail.

References

[1] I. Aleinov and E.G. Puckett. Computing surface tension with high-order kernels. In *Proc.* 6^{th} *International Symposium on Computational Dynamics, Lake Taho, CA*, Sep. 4-8, 1995.

[2] W. Aniszewski. Improvements, testing and development of the ADM-τ sub-grid surface tension model for two-phase LES. *J. Comp. Phys.*, 327: 389–415, 2016.

[3] W. Aniszewski, A. Boguslawski, M. Marek, and A. Tyliszczak. A new approach to sub-grid surface tension for LES of two-phase flows. *J. Comp. Phys.*, 231:7368–7397, 2012.

[4] S.V. Apte, M. Gorokhovski, and P. Moin. LES of atomizing spray with stochastic modeling of secondary breakup. *Int. J. Multiphase Flow*, 29: 1503–1522, June 2003.

[5] R. Bermejo and J. L. Prieto. A semi-Lagrangian particle level set finite element method for interface problems. *SIAM J. Sci. Comput.*, 35(4): A1815–A1846, 2013.

[6] E. Berrocal, E. Kristensson, P. Hottenbach, M. Aldén, and G. Grünefeld. Quantitative imaging of a non-combusting diesel spray using structured laser illumination planar imaging. *Appl. Phys. B*, 109(4):683–694, 2012.

[7] G. M. Bianchi, F. Minelli, R. Scardovelli, and S. Zaleski. 3D Large Scale Simulation of the High-Speed Liquid Jet Atomization. *SAE Technical Paper*, 01(0244), 2007.

[8] G.M. Bianchi, P. Pelloni, S. Toninel, R. Scardovelli, A. Leboissetier, and S. Zaleski. Improving the Knowledge of High-Speed Liquid Jets Atomization By Using Quasi-Direct 3D Simulation. *SAE Technical Paper*, 24 (089), 2005.

[9] G. Bornia, A. Cervone, S. Manservisi, R. Scardovelli, and S. Zaleski. On the properties and limitations of the height function method in two-dimensional Cartesian geometry. *J. Comp. Phys.*, 230:851–862, 2011.

[10] J. U. Brackbill. The Ringing Instability in Particle-in-Cell Calculations of Low-Speed Flow. *J. Comp. Phys.*, 75:469–492, 1988.

[11] J. U. Brackbill. Particle methods. *Int. J. Numer. Meth. Fl.*, 47:693–705, 2005.

[12] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comp. Phys.*, 100(2):335–354, 1992.

[13] A. Cervone, S. Manservisi, and R. Scardovelli. An optimal constrained approach for divergence-free velocity interpolation and multilevel VOF method. *Comput. Fluids*, 47:101–114, 2011.

[14] J. Chesnel, J. Reveillon, F.X. Demoulin, and T. Menard. Subgrid Analysis of Liquid Jet Atomization. *Atomization Spray*, 21(1):41–67, 2011.

[15] M. Chrigui, A.R. Masri, A. Sadiki, and J. Janicka. Large eddy simulation of a polydisperse ethanol spray flame. *Flow Turbulence Bombust*, 90:813–832, 2013.

[16] P.W. Cleary and J.J. Monaghan. Conduction Modelling Using Smoothed Particle Hydrodynamics. *J. Comp. Phys.*, 148:227–264, 1999.

[17] N.O. Crossland, S.W. Shires, and D. Bennett. Aquatic toxicology of cypermethrin. III. fate and biological effects of spray drift deposits in fresh water adjacent to agricultural land. *Aquat. Toxicol.*, 2:253–270, 1982.

[18] S.J. Cummins, M.M. Francois, and D.B. Kothe. Estimating curvature from volume fractions. *Comput. Struct.*, 83:425–434, 2005.

[19] E. De Villiers, AD Gosman, and HG Weller. Large Eddy Simulation of Primary Diesel Spray Atomization. *SAE Technical Paper*, 01(0100), 2004.

[20] O. Desjardins and H. Pitsch. A spectrally refined interface approach for simulating multiphase flows. *J. Comp. Phys.*, 20:1658–1677, 2009.

[21] O. Desjardins, V. Moureau, and H. Pitsch. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *J. Comp. Phys.*, 227(18):8395–8416, 2008.

[22] O. Desjardins, J.O. McCaslin, M. Owkes, and P. Brady. Direct numerical and large-eddy simulation of primary atomization in complex geometries. *Atomization Sprays*, 23(11):1001–1048, 2013.

[23] R.G. Dorman. The atomization of a liquid in a flat spray. *Br. J. App. Phys.*, 3:189–192, 1952.

[24] T.D. Dreeben and S.B. Pope. Probability density function and reynolds-stress modeling of near-wall turbulent flows. *Phys. Fluids*, 9(1):154–163, January 1997.

[25] B. Duret, J. Reveillon, T. Menard, and F.X. Demoulin. Improving primary atomization modeling through DNS of two-phase flows. *Int. J. Multiphase Flow*, 55:130–137, 2013.

[26] E. Edwards and R. Bridson. A high-order accurate particle-in-cell method. *Int. J. Numer. Meth. Engng.*, 90:1073–1088, 2012.

[27] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comp. Phys.*, 183 (1):83–116, 2002.

[28] D. Enright, F. Losasso, and R. Fedkiw. A fast and accurate semi-Lagrangian particle level set method. *Comput. Struct.*, 83:479–490, 2005.

[29] G.M. Faeth. Structure and atomization properties of dense turbulent sprays. *Twenty-Third Symposium (International) on Combustion/The Combustion Institute*, pages 1345–1352, 1990.

[30] G.M. Faeth, L.-P. Hsaing, and P.-K. Wu. Structure and breakup properties of sprays. *Int. J. Multiphase Flow*, 21:99–127, 1995.

[31] T.D. Fansler and S.E. Parrish. Spray measurement technology: a review. *Meas. Sci. Technol.*, 26:01002, 2015.

[32] R. P. Fedkiw, T. Aslam, B. Merriam, and S. Osher. A non-oscillatiory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comp. Phys.*, 152:457–492, 1999.

[33] R.O. Fox. Large-Eddy-Simulation tools for multiphase flows. *Annu. Rev. Fluid Mech.*, 44:47–76, 2012.

[34] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Siciliand, and M.W. Williams. A balanced-force algorithm for continuous and sharp

interfacial surface tension models within a volume tracking framework. *J. Comp. Phys.*, 213:141–173, 2006.

[35] D. Fuster, A. Bague, T. Boeck, L.L. Moyne, A. Leboissetier, S. Popinet, P. Ray, R. Scardovelli, and S. Zaleski. Simulation of primary atomization with an octree adaptive mesh refinement and VOF method. *Int. J. Multiphase Flow*, 35:550–565, 2009.

[36] E. Giffen and A. Muraszew. *The Atomization of Liquid Fuels*. Chapman and Hall, London, 1953.

[37] J. P. Gois, A. Nakano, L. G. Nonato, and G. C. Buscaglia. Front tracking with moving-least-squares surfaces. *J. Comp. Phys.*, 227:9643–9669, 2008.

[38] R. Goldman. Curvature formulas for implicit curves and surfaces. *Comput. Aided Geom. D.*, 22:632–658, 2005.

[39] C. Gong, M. Jangi, T. Lucchini, G. D'Errico, and X.-S. Bai. Large eddy simulation of air entrainment and mixing in reacting and non-reacting diesel sprays. *Flow Turbulence Combust.*, 93:385–404, 2014.

[40] M. Gorokhovski and M. Herrmann. Modeling primary atomization. *Annu. Rev. Fluid Mech.*, 40:343–366, 2008.

[41] M.A. Gorokhovski and V.L. Saveliev. Analysis of Kolmogorov's model of breakup and its application to Lagrangain computation of liquid sprays under air-blast atomization. *Phys. Fluids*, 15(1):184–191, 2003.

[42] H. Gotoh and A. Khayyer. On the state-of-the-art of particle methods for coastal and ocean engineering. *Coast. Eng. J.*, 60:79–103, 2018.

[43] F. Ham, S. Apte, G. Iaccarino, X. Wu, M. Herrmann, G. Constantinescu, K. Mahesh, and P. Moin. Unstructured LES of reacting multiphase flows

in realistic gas turbine combustors. Technical report, Center for Turbulence Research, 2003.

[44] F. H. Harlow. The particle-in-cell computing method for fluid dynamics. *Methods. Comput. Phys.*, 3:319–343, 1963.

[45] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *Phys. Fluids*, 8: 2182–2189, 1965.

[46] D.C. Haworth and S.B. Pope. A generalized langevin model for turbulent flows. *Phys. Fluids*, 29(2):387–405, 1986.

[47] J. Hernández, J. López, P. Gómez, C. Zanzi, and F. Faura. A new volume of fluid method in three dimensions – part I: multidimensional advection method with face-matched flux polyhedra. *Int. J. Numer. Methods Fluids*, 58(8):897–921, 2008.

[48] M. Herrmann. A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *J. Comp. Phys.*, 227(4): 2674–2706, 2008.

[49] M. Herrmann. A surface tension sub-grid model for phase interface dynamics. Technical report, Center for Turbulence Research, 2010.

[50] M. Herrmann. A parallel Eulerian interface tracking/Lagrangian point particle multi-scale coupling procedure. *J. Comp. Phys.*, 229:745–759, 2010.

[51] M. Herrmann. Detailed numerical simulations of the primary atomization of a turbulent liquid jet in crossflow. *J. Eng. Gas Turb. Power*, 132: 061506, 2010.

[52] M. Herrmann. On simulating primary atomization using the refined level set grid method. *Atomization Sprays*, 21(4):283–301, 2011.

[53] M. Herrmann. A dual scale volume-of-fluid approach for modeling turbulent phase interface dynamics. Technical report, Stanford Univeristy, 2014.

[54] M. Herrmann and M. Gorokhovski. An Outline of an LES Subgrid Model for Liquid/Gas Phase Interface Dynamics. *Proceedings of the 2008 CTR Summer Program,Center for Turbulence Research, Stanford University,CA*, pages 171–181, 2008.

[55] M. Herrmann and M. Gorokhovski. A large eddy simulation subgrid model for turbulent phase interface dynamics. 11th Triennial International Annual Conference on Liquid Atomization and Spray Systems, July 2009.

[56] S. E. Hieber and P. Koumoutsakos. A Lagrangian particle level set method. *J. Comp. Phys.*, 210(1):342–367, November 2005.

[57] J.O. Hinze. Fundamentals of the hydrodynamic mechanism of splitting in dispersion processes. *AIChE J.*, 1(3):289–295, 1955.

[58] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comp. Phys.*, 39(1):201–225, 1981.

[59] A.C. Iannetti and J.P. Moder. Comparing spray characteristics from Reynolds averaged Navier-Stokes (RANS) National Combustion Code (ncc) calculations against experimental data for a turbulent reacting flow. 48th Aerospace Sciences Meeting, AIAA–2010–578, January 2010.

[60] A. Irannejad and F. A. Jaberi. Large eddy simulation of turbulent spray breakup and evaporation. *Int. J. Multiphase Flow*, 61:108–128, 2014.

[61] A. Irannejad and F. A. Jaberi. Numerical study of high speed evaporating sprays. *Int. J. Multiphase Flow*, 70:58–76, 2015.

[62] A. Irannejad, A. Banaeizadeh, and F. Jaberi. Large eddy simulation of turbulent spray combustion. *Combust. Flame*, 162:431–450, 2015.

[63] M. Jain, R.S. Prakash, G. Tomar, and R.V. Ravikrishna. Secondary breakup of a drop at moderate Weber numbers. *Proc. Royal Soc. A*, 471:20140930, 2015.

[64] A. Kastengren, J. Ilavsky, J.P. Viera, R. Payri, D.J. Duke, A. Swantek, F.Z. Tilocco, N. Sovis, and C.F. Powell. Measurements of droplet size in shear-driven atomization using ultra-small angle x-ray scattering. *Int. J. Multiphase Flow*, 92:131–139, 2017.

[65] D. Kedelty, J. Uglietta, and M. Herrmann. A Volume of Fluid Dual Scale Approach for Modeling Turbulent Liquid/Gas Phase Interfaces. In *ILASS-Americas* 29^{th} *Annual Conference, Atlanta, GA, USA*, May 2017.

[66] A. Khayyer and H. Gotoh. Enhancement of performance and stability of MPS mesh-free particle method for multiphase flows characterized by high density ratios. *J. Comp. Phys.*, 242:211–233, 2013.

[67] A. Khayyer, H. Gotoh, and Y. Shimizu. Comparative study on accuracy and conservation properties of two particle regularization schemes and proposal of an optimized particle shifting scheme in ISPH context. *J. Comp. Phys.*, 332:236–256, 2017.

[68] A. Khayyer, H. Gotoh, and Y. Shimizu. A projection-based particle method with optimized particle shifting for multiphase flows with large density ratios and discontinuous density fields. *Comput. Fluids*, 179:356–371, 2019.

[69] S. Koshizuka and Y. Oka. Moving particle semi-implicit method for fragmentation of incompressible fluid. *Nucl. Sci. Eng.*, 123:421–434, 1996.

[70] E. Kristensson, L. Araneo, E. Berrocal, J. Manin, M. Richter, M Aldén, and M. Linne. Analysis of multiple scattering suppression using structured laster illumination planar imaging in scattering and fluorescing media. *Opt. Express*, 19(14):13647–13663, 2011.

[71] R. Kurose, O. Desjardins, M. Nakamura, F. Akamatsu, and H. Pitsch. Numerical simulations of spray flames. Technical report, Center for Turbulence Research, 2004.

[72] J. Larocque, S. Vincent, D. Lacanette, P. Lubin, and J.-P. Caltagirone. Parametric study of LES subgrid terms in a turbulent phase separation flow. *Int. J. Heat Fluid Fl.*, 31:536–544, 2010.

[73] R. Lebas, T. Menard, P.A. Beau, A. Berlemont, and F.X. Demoulin. Numerical simulation of primary break-up and atomization: DNS and modelling study. *Int. J. Multiphase Flow*, 35:247–260, 2008.

[74] T. Lederlin and H. Pitsch. Large-eddy simulation of an evaporating and reacting spray. Technical report, Center for Turbulence Research, 2008.

[75] H. Lei, C. J. Mundy, and G. Schenter. Modeling nanoscale hydrodynamics by smoothed dissipative particle dynamics. *J. Chem. Phys.*, 142:194504, 2015.

[76] R.J. Leveque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.*, 33(2):627–665, 1996.

[77] J. Li, Y. Y. Renardy, and M. Renardy. Numerical simulation of breakup

of a viscous drop in simple shear flow through a volume-of-fluid method. *Phys. Fluids*, 12(2):269–282, February 2000. doi: 10.1063/1.870305.

[78] Z. Li, F. A. Jaberi, and T. I-P. Shih. A hybrid Lagrangian-Eulerian particle-level set method for numerical simulations of two-fluid turbulent flows. *Int. J. Numer. Meth. Fluids*, 56:2271–2300, 2008.

[79] S.J. Lind, R. Xu, P.K. Stansby, and B.D. Rogers. Incompressible smoothed particle hydrodynamics for free-surface flows: a generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *J. Comp. Phys.*, 231(4):1499–1523, 2012.

[80] M. Linne. Imaging in the optically dense regions of a spray: A review of developing techniques. *Prog. Energy Combust. Sci.*, 39(5):403–440, 2013.

[81] M. Linne, M. Paciaroni, T. Hall, and T. Parker. Ballistic imaging of the near field in a diesel spray. *Exp. Fluids*, 40:836–846, 2006.

[82] J. Liu, S. Koshizuka, and Y. Oka. A hybrid particle-mesh method for viscous, incompressible, multiphase flows. *J. Comp. Phys.*, pages 65–93, 2005.

[83] M. B. Liu and G. R. Liu. Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Arch. Comput. Methods Engrg.*, 17(1): 25–76, 2010.

[84] M. B. Liu, W. P. Xie, and G. R. Liu. Modeling incompressible flows using a finite particle method. *Appl. Math. Model.*, 29(12):1252–1270, 2005.

[85] W. Liu. *Advanced simulation and modeling of turbulent sprays.* PhD thesis, University of Minnesota, March 2014.

[86] J. López, J. Hernández, P. Gómez, and F. Faura. A volume of fluid method based on multidimensional advection and spline interface reconstruction. *J. Comp. Phys.*, 195(2):718–742, 2004.

[87] L.B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astron. J.*, 82:1013–1024, 1977.

[88] P. Macklin and J. Lowengrub. An improved geometry-aware curvature discretization for level set methods: application to tumor growth. *J. Comp. Phys.*, 215:392–401, 2006.

[89] K. Mahesh, G. Constantinescu, S. Apte, G. Iaccarino, and P. Moin. Large-eddy simulation of gas turbine combustors. Technical report, Center for Turbulence Research, 2001.

[90] S. Marrone, A. Di Mascio, and D. Le Touze. Coupling of smoothed particle hydrodynamics with finite volume method for free-surface flows. *J. Comp. Phys.*, 310:161–180, 2016.

[91] W.O.H. Mayer and R. Branam. Atomization characteristics on the surface of a round liquid jet. *Exp. in Fluids*, 36:528–539, 2004.

[92] T. Menard, S. Tanguy, and A. Berlemont. Coupling level set/VOF/ghost fluid methods: validation and application to 3D simulation of the primary break-up of a liquid jet. *Int. J. Multiphase Flow*, 33(5):510–524, 2007.

[93] J.J. Monaghan. Smoothed particle hydrodynamics. *Ann. Rev. Astron. Astrophys.*, 30:543–574, 1992.

[94] J.J. Monaghan and A. Rafiee. A simple SPH algorithm for multi-fluid flow with high density ratios. *Int. J. Numer. Meth. Fluids*, 71:537–561, 2013.

[95] G. Moretti. Computation of flows with shocks. *Ann. Rev. Fluid Mech.*, 19:313–337, 1987.

[96] J. P. Morris. Simulating surface tension with smoothed particle hydrodynamics. *Int. J. Numer. Meth. Fl.*, 33:333–353, 2000.

[97] B.D. Nichols, C.W. Hirt, and R.S. Hotchkiss. SOLA-VOF: A solution algorithm for transient fluid flow with multiple free boundaries, LA-8355. Technical report, Los Alamos National Laboratory, 1980.

[98] N. Nishio, K. Yamana, Y. Yamaguchi, T. Inaba, K. Kuroda, T. Nakajima, K. Ohno, and H. Fujimura. Large-scale SPH simulations of droplet impact onto a liquid surface up to the consequent formation of Worthington jet. *Int. J. Numer. Meth. Fl.*, 63(12):1435–1447, 2010.

[99] K. Okuyama and I.W. Lenggoro. Preparation of nanoparticles via spray route. *Chem. Eng. Sci.*, 58:537–547, 2003.

[100] E. Olsson and G. Kreiss. A conservative level set method for two phase flow. *J. Comp. Phys.*, 210(1):225–246, 2005.

[101] E. Olsson, G. Kreiss, and S. Zahedi. A conservative level set method for two phase flow II. *J. Comp. Phys.*, 225(1):785–807, 2007.

[102] P.J. O'Rourke and A.A. Amsden. The TAB method for numerical calculation of spray droplet breakup. *SAE Paper 872089*, 1987.

[103] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comp. Phys.*, 79(1):12–49, 1988.

[104] M. Owkes and O. Desjardins. A computational framework for conservative, three-dimensional, unsplit, geometric transport with application to the Volume-of-Fluid (VoF) method. *J. Comp. Phys.*, 270:587–612, 2014.

[105] M. Owkes and O. Desjardins. A mesh-decoupled height function method for computing interface curvature. *J. Comp. Phys.*, 281:285–300, 2015.

[106] M. Owkes and O. Desjardins. A mass and momentum conserving unsplit semi-Lagrangian framework for simulating multiphase flows. *J. Comp. Phys.*, 332:21–46, 2017.

[107] M. Owkes, E. Cauble, J. Senecal, and R. A. Currie. Importance of curvature evaluation scale for predictive simulations of dynamic gas-liquid interfaces. *J. Comp. Phys.*, 365:37–55, 2018.

[108] M.G. Pai, O. Desjardins, and H. Pitsch. Detailed simulations of primary breakup of turbulent liquid jets in crossflow. Technical report, Center for Turbulence Research, 2008.

[109] M.G. Pai, I. Bermejo-Moreno, O. Desjardins, and H. Pitsch. Role of Weber number in primary breakup of turbulent liquid jets in crossflow. Technical report, Center for Turbulence Research, 2009.

[110] S.B. Pope. Ten questions concerning the large-eddy simulation of turbulent flows. *New J. Phys.*, 6(35):2–23, 2003.

[111] S.B. Pope. Simple models of turbulent flows. *Phys. Fluids*, 23:011301, 2011.

[112] S. Popinet. An accurate adaptive solver for surface-tension-driven interfacial flows. *J. Comp. Phys.*, 228:5838–5866, 2009.

[113] D. J. Price. Smoothed particle hydrodynamics and magnetohydrodynamics. *J. Comp. Phys.*, 227(3):759–794, 2012.

[114] M. Rahm, M. Paciaroni, Z. Wang, D. Sedarsky, and M. Linne. Evaluation of optical arrangements for ballistic imaging in sprays. *Opt. Express*, 23 (17):243730, 2015.

[115] R.D. Reitz. Modeling atomization processes in high-pressure vaporizing sprays. *Atomiz. Spray Technol.*, 3:309–337, 1987.

[116] Y. Renardy and M. Renardy. PROST: A Parabolic Reconstruction of Surface Tension for the Volume-of-Fluid Method. *J. Comp. Phys.*, 183: 400–421, 2002.

[117] M. Rudman. A volume-tracking method for incompressible multifluid flows with large density variations. *Int. J. Numer. Meth. Fl.*, 28:357–378, 1998.

[118] S.S. Sazhin, S.B. Martynov, T. Kristyadi, C. Crua, and M.R. Heikal. Diesel fuel spray penetration, heating, evaporation and ignition: modelling vs. experimentation. *Int. J. Eng. Syst. Model. Simul.*, 1(1):1–19, 2008.

[119] R. Scardovelli and S. Zaleski. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *J. Comp. Phys.*, 164: 228–237, 2000.

[120] C. Schroeder, W. Zheng, and R. Fedkiw. Semi-implicit surface tension formulation with a Lagrangian surface mesh on an Eulerian simulation grid. *J. Comp. Phys.*, 231(4):2092–2115, 2012.

[121] A. Scotti and C. Meneveau. A fractal model for large eddy simulation of turbulent flow. *Physica D*, 127:198–232, 1999.

[122] P.K. Senecal, D.P. Schmidt, I. Nouar, C.J. Rutland, R.D. Reitz, and M.L. Corradini. Modeling high-speed viscous liquid sheet atomization. *Int. J. Multiphase Flow*, 25:1073–1097, 1999.

[123] J. Shinjo. Recent Advances in Computational Modeling of Primary Atomization of Liquid Fuel Sprays. *Energies*, 11:2971, 2018.

[124] J. Shinjo and A. Umemura. Simulation of liquid jet primary breakup: dynamics of ligament and droplet formation. *Int. J. Multiphase Flow*, 36 (7):513–532, 2010.

[125] F. V. Sirotkin and J. J. Yoh. A new particle method for simulating breakup of liquid jets. *J. Comp. Phys.*, 231(4):1650–1674, February 2012.

[126] V. Srinivasan, A.J. Salazar, and K. Saito. Numerical Investigation on the Disintegration of Round Turbulent Liquid Jets Using LES/VOF Techniques. *Atomization Spray*, 18(7):571–618, 2008.

[127] H.A. Stone. Dynamics of drop deformation and breakup in viscous fluids. *Annu. Rev. Fluid Mech.*, 26:65–102, 1994.

[128] M. Sussman and M. Ohta. High-Order Techniques for Calculating Surface Tension Forces. *J. Comp. Phys.*, 154:425–434, 2006.

[129] M. Sussman and M. Ohta. Improvements for calculating two-phase bubble and drop motion using an adaptive sharp interface method. *Fluid Dyn. Mater. Proc.*, 3(1):21–36, 2007.

[130] M. Sussman and E. G. Puckett. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *J. Comp. Phys.*, 162(2):301–337, 2000.

[131] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Phys. Fluids A*, 114(1):146–159, 1994.

[132] A. M. Tartakovsky and P. Meakin. Pore scale modeling of immiscible and miscible fluid flows using smoothed particle hydrodynamics. *Adv. Water Resour.*, 29:1464–1478, 2006.

[133] A. M. Tartakovsky and A. Panchenko. Pairwise force smoothed particle hydrodynamics model for multiphase flow: surface tension and contact line dynamics. *J. Comp. Phys.*, 305:1119–1146, 2016.

[134] H. Terashima and G. Tryggvason. A front-tracking/ghost-fluid method for fluid interfaces in compressible flows. *J. Comp. Phys.*, 228:4012–4037, 2009.

[135] X.S. Tian, H. Zhao, H.F. Liu, W.F. Li, and J.L. Xu. Three-dimensional large eddy simulation of round liquid jet primary breakup in coaxial gas flow using the vof method. *Fuel Process Technol.*, 131:396–402, 2015.

[136] D. J. Torres and J. U. Brackbill. The point-set method: front-tracking without connectivity. *J. Comp. Phys.*, 165:620–644, 2000.

[137] N. Tsuruta, A. Khayyer, and H. Gotoh. A short note on dynamic stabilization of moving particle semi-implicit method. *Comput. Fluids*, 82: 158–164, 2013.

[138] M. R. Turner, S. S. Sazhin, J. J. Healey, C. Crua, and S. B. Martynov. A breakup model for transient diesel fuel sprays. *Fuel*, 97:288–305, 2012.

[139] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comp. Phys.*, 100:25–37, 1992.

[140] C.M. Vargas, J.C. Lasheras, and E.J. Hopfinger. Initial breakup of a small-diamter liquid jet by a high-speed gas stream. *J. Fluid. Mech.*, 497: 405–434, 2003.

[141] R. Vehring. Pharmaceutical particle engineering via spray drying. *Pharm. Res.*, 25(5):999–1022, 2008.

[142] R. Vehring, W.R. Foss, and D. Lechuga-Ballesteros. Particle formation in spray drying. *J. Aerosol Sci.*, 38:728–746, 2007.

[143] F. Wang, J. Wu, and Z. Liu. Surface Tensions of Mixtures of Diesel Oil or Gasoline and Dimethoxymethane, Dimethyl Carbonate, or Ethanol. *Energy Fuels*, 20(6):2471–2474, 2006.

[144] H. Wang, R. Agrusta, and J. van Hunen. Advantages of a conservative velocity interpolation (CVI) scheme for particle-in-cell methods with application in geodynamic modeling. *Geochem. Geophys.*, (10.1002/2015GC005824):2015–2023, 2015.

[145] L. Wang, A. Khayyer, H. Gotoh, Q. Jiang, and C. Zhang. Enhancement of pressure calculation in projection-based particle methods by incorporation of background mesh scheme. *Appl. Ocean Res.*, 86:320–339, 2019.

[146] E. A. Wenzel and S. C. Garrick. Modeling and Simulation of Droplet Dynamics with a Lagrangian Volume of Fluid method. In *Proceedings of*

the 2^{nd} *Thermal and Fluid Engineering Conference*, Las Vegas, Nevada, USA, 2017.

[147] E. A. Wenzel and S. C. Garrick. A finite particle approach for the simulation of multiphase flows. In *Proceedings of the 29th Annual Conference on Liquid Atomization and Spray Systems*, Atlanta, GA, USA, 2017.

[148] E. A. Wenzel and S. C. Garrick. Finite particle methods for computing interfacial curvature in volume of fluid simulations. *At. Sprays*, 28(2): 141–160, 2018.

[149] E. A. Wenzel and S. C. Garrick. Dynamic evaluation of the finite particle method for computing surface tension. In *Proceedings of the 14th International Conference on Liquid Atomization and Spray Systems*, Chicaco, IL, USA, 2018.

[150] E. A. Wenzel, F. A. Kulacki, and S. C. Garrick. Modeling and simulation of liquid-liquid droplet heating in a laminar boundary layer. *Int. J. Heat Mass Trans.*, 97:653–661, 2016.

[151] E.A. Wenzel and S.C. Garrick. A point-mass particle method for the simulation of immiscible multiphase flows on an Eulerian grid. *J. Comp. Phys.*, 387:108835, November 2019.

[152] G.D. Weymouth and D.K.P. Yue. Conservative volume-of-fluid method for free-surface simulations on cartesian-grids. *J. Comp. Phys.*, 229(8): 2853–2865, 2010.

[153] M.W. Williams, B.B. Kothe, and E.G. Puckett. Convergence and Accuracy of Kernel-Based Continuum Surface Tension Models. In 13^{th} *US National Congress of Applied Mechanics, Gainesville, FL, USA*, 1998.

[154] F. Xiao, M. Dianat, and J.J. McGurik. LES of turbulent liquid jet primary breakup in turbulent coaxial air flow. *Int. J. Multiphase Flow*, 60:103–118, 2014.

[155] F. Xiao, M. Dianat, and J.J. McGurik. Large eddy simulation of a single droplet and liquid jet primary breakup using a coupled level set/volume of fluid method. *Atomization Sprays*, 24(4):281–302, 2014.

[156] R. Xu, P.K. Stansby, and D. Laurence. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *J. Comp. Phys.*, 228(18):6703–6725, 2009.

[157] H. Zhang, B. Bai, and Y. Wang. Quantitative description of droplet dispersion of hollow cone spray in gaseous crossflow. *Exp. Therm. Fluid Sci.*, 93:398–408, 2018.

[158] W. Zheng, B. Zhu, B. Kim, and R. Fedkiw. A new incompressibility discretization for a hybrid particle MAC grid representation with surface tension. *J. Comp. Phys.*, 280:96–142, 2015.

[159] L. Zhou, J. Xia, J. Shinjo, A. Cairns, L. Cruff, and H. Blaxill. Towards high-fidelity multi-scale simulation of spray atomization. *Energy Procedia*, 66:309–312, 2015.

[160] C. Zhu, M. Ertl, and B. Weigand. Numerical investigation on the primary breakup of an inelastic non-Newtonian liquid jet with inflow turbulence. *Phys. Fluids*, 25:083102, 2013.

[161] D. Zuzio, J.L. Estivalezes, P. Villedieu, and G. Blanchard. Numerical simulation of primary and secondary atomization. *C. R. Macanique*, 341: 15–25, 2013.